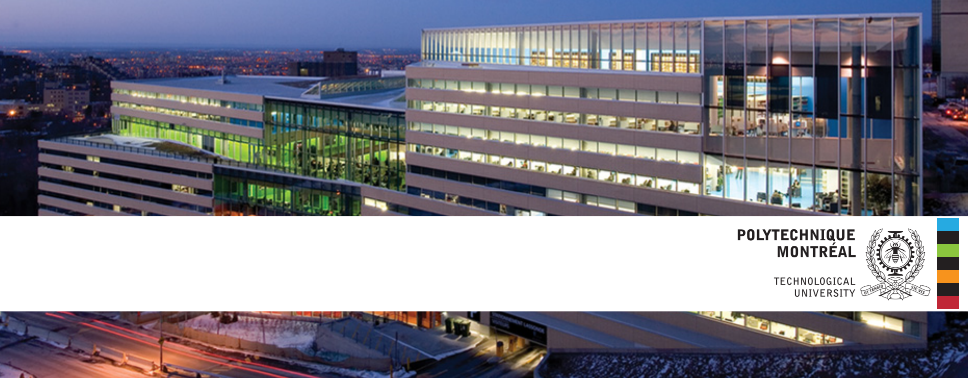
A Case Study For Using Chapel Within The Global Aerospace Industry

The 7th Annual Parallel Applications Workshop, Alternatives To MPI+X

Éric Laurendeau, PhD, Professor
Karim Mohamad Zayni, PhD Student

November 17, 2024



POLYTECHNIQUE MONTRÉAL
TECHNOLOGICAL UNIVERSITY

Table of Contents

Aerospace Global Objectives

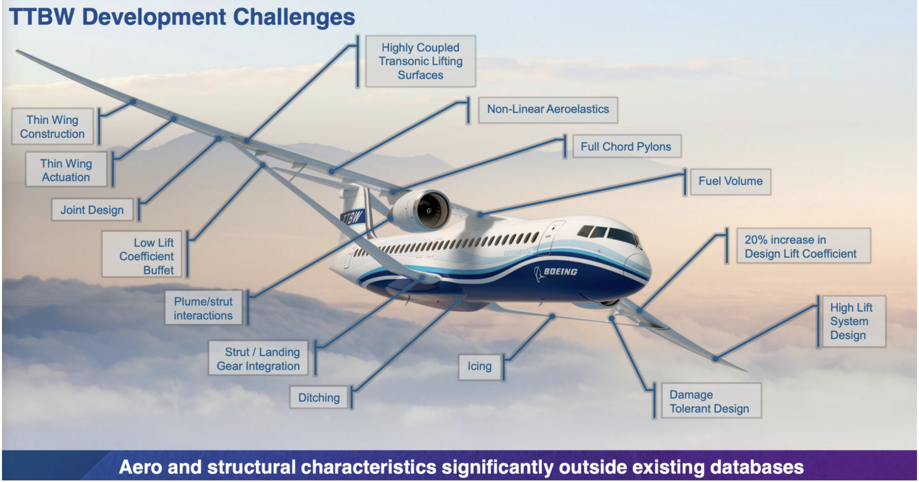
Global Warming: Zero net carbon emissions by 2050

- Sustainable Aviation Fuel (SAF)
- Hydrogen, Electrical, Hybrid propulsion
- Novel Aircraft Configurations

NASA
Transonic Truss-Braced Wing

Bombardier
Ecojet

NASA TTBW: Aerodynamics proves Key!

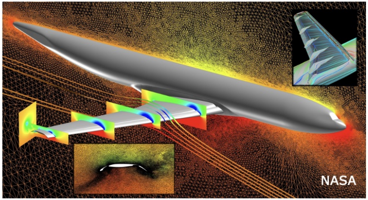


Source: NASA/Boeing

Aerodynamics: Available Tools

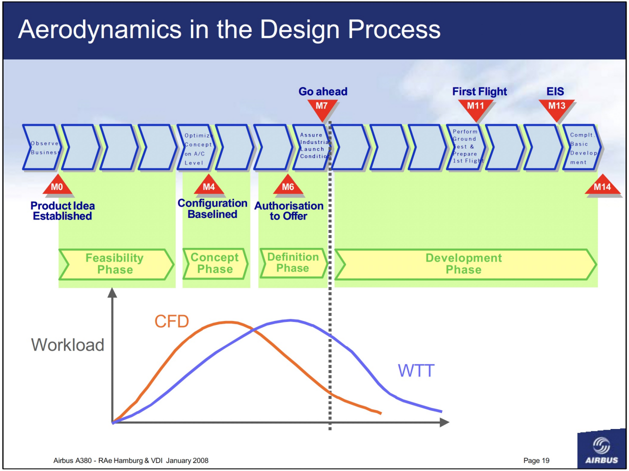
Flight tests ~100 000$/hr
Prototype: 1 Billion$

Wind-tunnel tests ~1000-10 000$/hr
Prototype: 10 Millions$



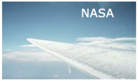
CFD ~100-1000$/hr
Prototype: 1 Million$

Aerodynamic within Design Process



Source: Axel Flaig, Royal Aeronautical Society, Jan. 2008

Aerodynamics: CFD Workflow (aero-icing)



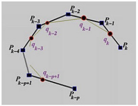
physics

$$\underbrace{\rho\left(\frac{\partial \vec{u}}{\partial t} + \vec{u}\cdot\nabla\vec{u}\right)}_{\text{Acceleration}} = \underbrace{-\nabla p}_{\text{Pressure}} + \underbrace{\nu\Delta\vec{u}}_{\text{Viscosity}}$$

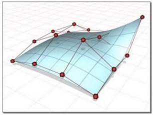
$$\underbrace{\nabla\cdot\vec{u} = 0}_{\text{Continuity Equation}}$$

MIT

Applied Math.

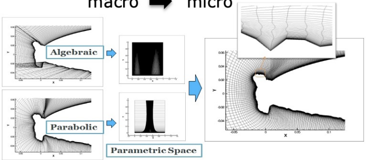


geometry parametrization (NURBS)

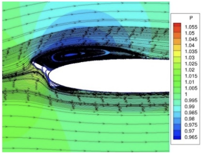


Geometry discretization(CAD)

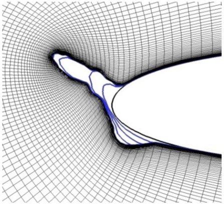
macro ➡ micro



Volumetric mesh generation
10-1000 Millions unknowns



solution:
i) flow, ii) droplets



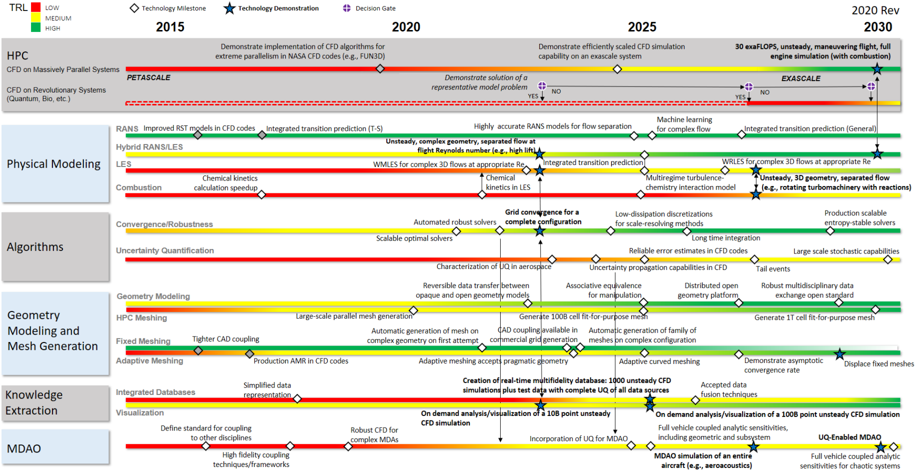
Time-advancement
& post-processing

```
#include <stdio.h>

int main()
{
    int i;
    int a[10];
    printf("Enter student's scores: \n");
    for(i = 0; i < 10; i++) {
        scanf("%d", &a[i]);
    }
    printf("Your student's scores are: \n\n");
    for(i = 0; i < 10; i++) {
        printf("%d\n", a[i]);
    }
    return 0;
}
```

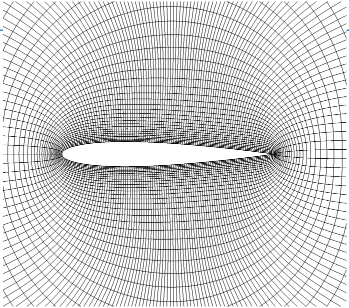
Source code

HPC

NASA Computational AeroScience Roadmap

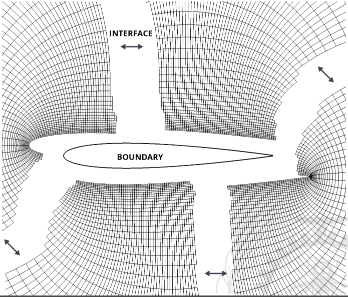


Parallel CFD for HPC



Solving Strategy

- **Volumetric Meshing** around complex geometries
- **2D Meshes**: Ranging from 0.5 to 1.0 Million Unknowns
- **3D Meshes**: Handling up to 1 Billion Unknowns
- **High-Performance Computing (HPC)**: Leveraged to significantly reduce computation time
- **Problem Decomposition**: The problem is partitioned into smaller sub-problems interconnected via interfaces
- **Task Parallelization**: Each sub-problem runs independently on dedicated tasks
- **Communication Optimization**: Minimizing communication overhead to maximize overall efficiency



CHAMPS: Advanced 2D-3D CFD Solver

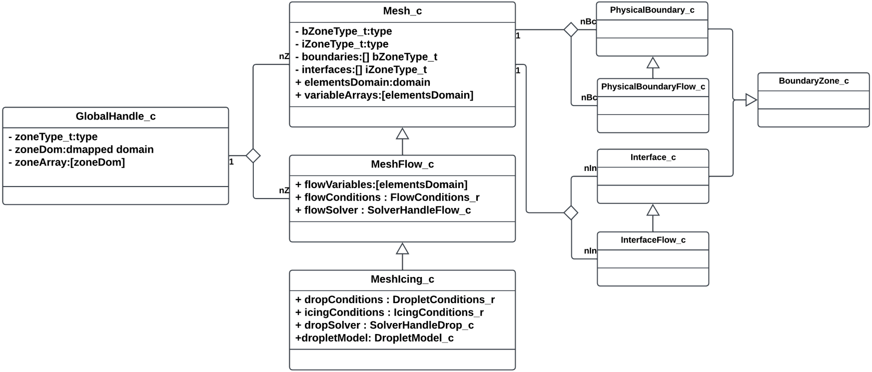
CHAMPS (CHapel MultiPhyics Software)

- 2D-3D Unstructured Reynolds Average Navier Stokes Solver
- Second order finite volume
- Convective Fluxes : Roe, AUSM
- SA, k-ω SST-V and Langtry-Menter transitional turbulence models
- Explicit solver (Runge-Kutta) and implicit solvers (SGS, GMRES)
- Linked to external libraries: MKL, CGNS, METIS and PETSC
- Multi-Fidelity Solvers: Potential, Non Linear Vortex Lattice, Euler, RANS
- Multi-Physics: Icing (Deterministic, Stochastic), Fluid-Structure Interactions

CHAMPS Key Stats

- Total number of code lines $\approx$ 150K lines
- Pre-Processor $\approx$ 17K lines
- Flow Solver $\approx$ 15K lines
- Turbulence Solver $\approx$ 13K lines
- Droplet Solver (Eulerian + Lagrangian) $\approx$ 24K lines
- Post Processor $\approx$ 2K lines

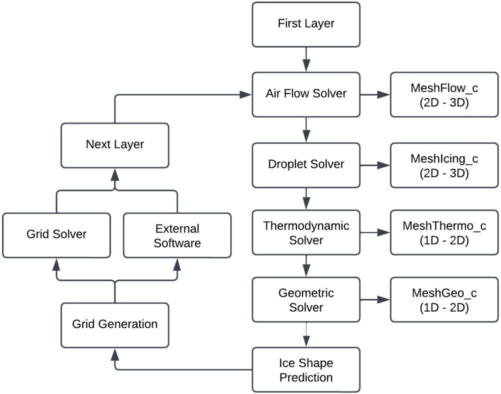
Software Structure



Software Overview

- Multiphysics problems require different computational domains grid
- *Type* aliases in *Chapel* are used to define computational domains at the start of each simulations
- Supports Generic Programming and Improve flexibility

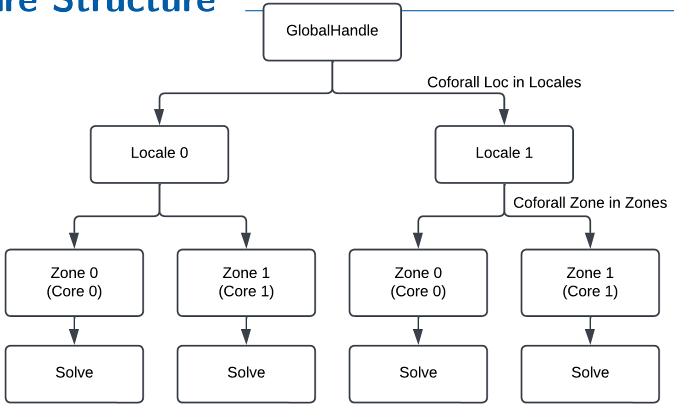
Software Structure



Example: Icing Framework

- Typical icing simulations involve four distinct computational domains
- Some domains solve the volume field (3D), while others focus on surface interactions (2D)
- Each computational domain has its own specific characteristics, variables and requirements

Software Structure



Software Overview

- Each simulation runs in a single execution (using a GlobalHandle), ensuring efficiency and consistency.
- Tasks are distributed hierarchically using *coforall* statements, first at the Locale (node) level, then further subdivided at the Zone (core) level to maximize parallelism.
- Grid partitioning through METIS guarantees optimal load balancing across all cores, enhancing computational efficiency and minimizing idle time.

```
proc main(args : [] string)
{
    // Fetching user inputs for the flow
    var flowInputs  : FlowInputs_r;
    var turbInputs  : TurbInputs_r;

    ref commonInputs = flowInputs.commonInputs_;

    var globalHandle : GlobalHandleFlow_t = initializeComputationalDomain(GlobalHandleFlow_t, commonInputs);

    try! writeln("Elapsed time after reading grid: ", clock.elapsed());

    runFlowSimulation(globalHandle.borrow(), clock, flowInputs, turbInputs);
}
```

Example: Flow Main

- Type `GlobalHandleFlow_t` will initialize `MeshFlow_t` computational domains
- One proc to run : `runFlowSimulation()`

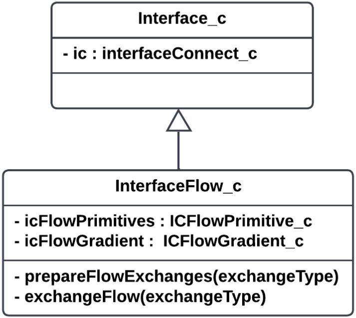
Software Structure

```
coforall loc in Locales do
on loc
{
    const localZonesIndices = globalHandle.zones_.localSubdomain();
    ref   localZones        = globalHandle.zones_.localSlice(localZonesIndices);

    coforall zone in localZones.borrow()
    {
        zone.flowModel_.solve(zone, locID, zone.localZoneIndex_);
    }
}
```

Example: Flow Main

- First `coforall` will loop over Locales
- Second `coforall` will loop over cores to execute `Solve()`

Software Structure

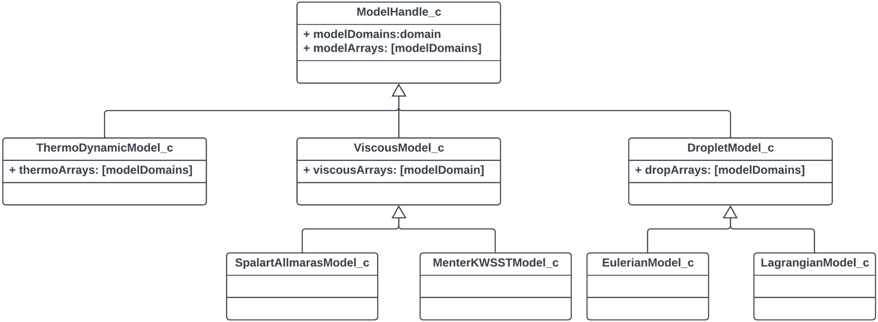



```
proc performInterfaceExchanges(zone, exchangeType : ExchangeType_t)
    {
        use CustomAllLocalesBarriers;

        // Fill buffers
        zone.prepareExchange(exchangeType);

        customAllLocalesBarrier.barrier();

        // Read buffers
        zone.exchangeInterfaces(exchangeType);

        customAllLocalesBarrier.barrier();
    }
```

Communication Overview

- Each `interfaceZone` is equipped with an `interfaceConnect` to facilitate seamless communication with adjacent zones.
- Each zone prepares for data exchanges by populating its respective buffer arrays, ensuring that all necessary information is readily available for transfer.
- Custom synchronization barriers are implemented to maximize efficiency.
- Once synchronization is achieved, all data exchanges are executed simultaneously, minimizing communication overhead and maximizing throughput.
- The global namespace support provided by *Chapel* ensures that any task can access the necessary buffers, regardless of its location across Locales.

Software Structure



Model Implementation Strategy

- All models inherit from a base ModelHandle_c Class
- Maximise code reusibility, leading to faster implementation and enhance redeability
- *Where* statements are needed to prevent compilation errors.
- This ensures compatibility when fields or methods are not present in the parent class.
- *Where* statements also prevent conflicts with sibling classes (other children of the same parent).

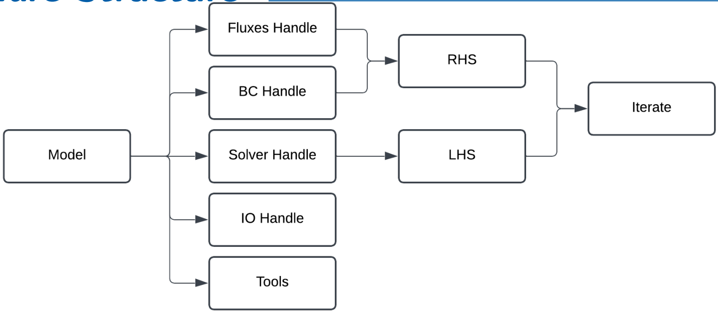
Software Structure

```
proc initializeViscousModelAndSolver(globalHandle, zone, flowInputs : FlowInputs_r, turbInputs : TurbInput
{
    select flowInputs.FLOW_REGIME_
    {
        when FlowRegime_t.INVISCID
        {
            zone.viscousModel_ = new owned InviscidModel_c();
        }
        when FlowRegime_t.LAMINAR
        {
            zone.viscousModel_ = new owned LaminarModel_c();
        }
        when FlowRegime_t.TURBULENT
        {
            select turbInputs.TURB_MODEL_
            {
                when TurbulenceModel_t.SA
                {
                    zone.viscousModel_ = new owned SpalartAllmarasModel_c(zone, turbInputs);
                }
                when TurbulenceModel_t.KW
                {
                    zone.viscousModel_ = new owned MenterKWSSTModel_c(zone, turbInputs);
                }
            }

            zone.viscousModel_.initializeConditionsAndSolvers(globalHandle, zone);
        }
    }
}
```

Example: Viscous Models

- Viscous models are decided based on user input at the start of `runFlowSimulation()`
- Leads to the instantiation of a new `viscousModel_c` object in each zone

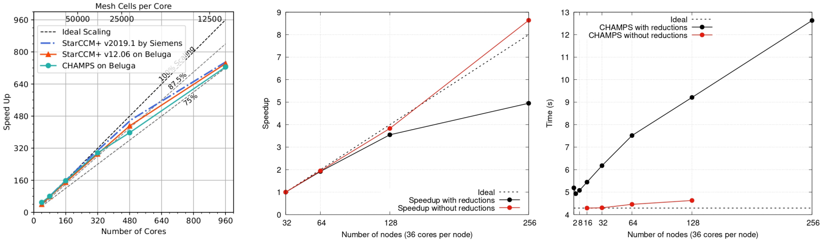
Software Structure



Model Implementation Strategy

- All models share a uniform structure, which simplifies implementation and the maintenance.
- Flux handlers populate the Right-Hand Side (RHS) of the equations based on the chosen convective flux scheme (e.g., Upwind, Roe, etc.), ensuring flexibility and adaptability to different numerical methods.
- Boundary condition handlers apply the appropriate boundary conditions directly to the RHS, ensuring accurate representation of physical constraints at the domain boundaries.
- The solver module constructs the Left-Hand Side (LHS) of the system and iterates towards the solution
- IO handlers manage user input and ensure proper output of simulation data
- Certain modules that require specific tools (e.g. geometric tools) are isolated into separate components to enhance code readability, modularity, and reusability across different parts of the project.

Software parallelization performance.



(a) Civil Airliner : Strong scaling *(b)* Simple box: Strong scaling *(c)* Simple box: Weak scaling

Parallel Scaling Efficiency of CHAMPS

Strong and weak scaling

To evaluate CHAMPS's efficiency in utilizing computational resources, two test cases were conducted to assess the code's strong and weak scaling performance.

- First case, civil airlines model: Speedup results, shown in Figure 1(a), are scaled from a 2-node baseline to highlight communication effects and are compared to the results obtained with the StarCCM+ software using the same mesh.
- Second test, plain box model: conducted on an HPE HPC cluster, used a high number of cores to explore CHAMPS scaling in detail. Figures 1(b) and 1(c) show strong and weak scaling results. Tests with and without reduction operations reveal CHAMPS' super scalability in the absence of reductions, indicating potential performance gains when I/O operations are performed "on-the-fly".

AIAA High Lift Prediction Workshop 5

AIAA High Lift Prediction

The HLPW5 is an international workshop focused on high-lift aerodynamic prediction. It brings together researchers, engineers, and experts to evaluate and improve CFD methods applied to high-lift configurations, such as aircraft wings with deployed slats and flaps during takeoff and landing. Its objectives are:

- Benchmarking CFD Methods
- Enhancing Accuracy of the CFD methods
- Promoting Collaboration within the community
- Guiding Future Development of CFD methods.

These simulations demand extremely fine meshes, with a number of cells increasing at each successive workshop edition.

TC1	TC2				TC3			
1.1	2.1	2.2	2.3	2.4	3.1	3.2	3.3	3.4
3M, 12M, 36M 78M, 144M, 240M 370M, 541M, 758M	5M 11M 59M 128M 394M	11M 26M 141M 301M 914M	14M 32M 168M 360M 1.1B	20M 44M 205M 429M 1.3B	52M 162M 352M 954M	58M 179M 407M 1.09B	62M 189M 444M 1.18B	64M 196M 456M 1.23B

AIAA High Lift Prediction Workshops

AIAA High Lift Prediction

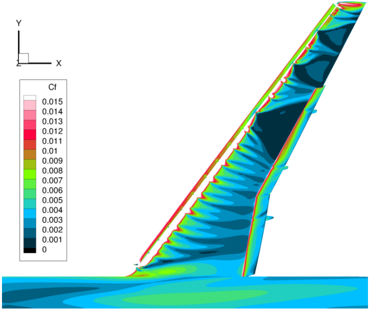
The High Lift prediction workshops have gained in popularity over the last decade and have greatly evolved:

- Increased number of participants from various academic and industrial teams, NASA, ONERA, JAXA, Bombardier, Airbus, Boeing, Polytechnique Montréal, and Politecnico Milano.
- The complexity of the simulations has increased, starting from 2D simulations to 3D full configurations aircraft.
- A recent increase in the number of simulations to be performed, along with the use of finer meshes, has been observed.
- Increased use of computational resources, from 576 cores for the first edition to 4608 cores during the last edition.

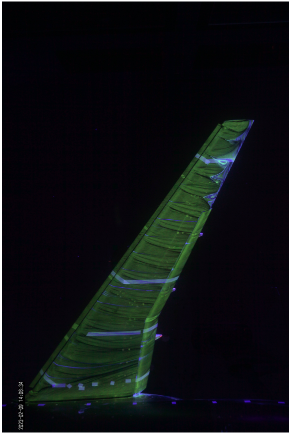
These workshops are very popular within the scientific community, serving as essential verification test cases. There's little doubt that future editions will be as popular, if not more, than the previous ones.

HLPW edition	Number of simulations to perform	Max mesh size (#cells)	Number of participants
1st-2010	20	161M	18
2nd-2013	19	545M	29
3rd-2017	21	565M	40
4th-2022	17	1.0B	57
5th-2024	39	1.3B	67

AIAA HLPW5 - pizza slices

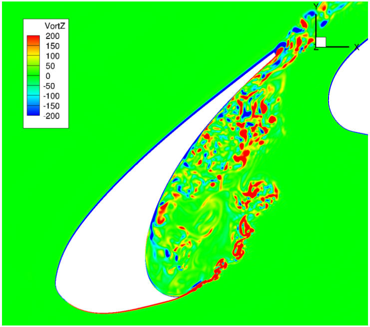


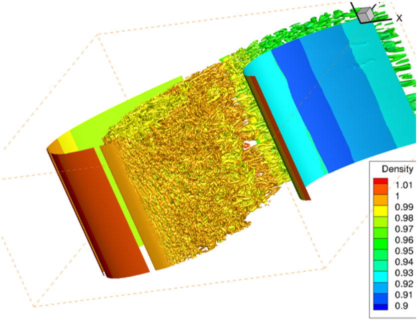
(a) $\alpha = 20.7^{\circ}$, TC2.3;



(b) $\alpha = 20.7^{\circ}$, Oil-flow visualization; [?]

Scale resolving simulations





30P30N turbulent slat cove

There is a growing trend in academia and industry toward scale-resolving simulations, which require computational meshes with hundreds of millions of cells and run over hundreds of thousands of time steps. This shift emphasizes the significant demand for computational resources within the CFD comunity.

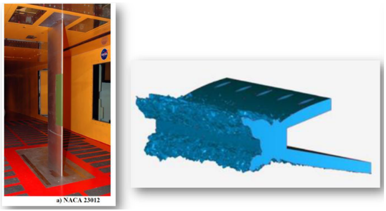
Ice Prediction Workshops

AIAA Ice Prediction

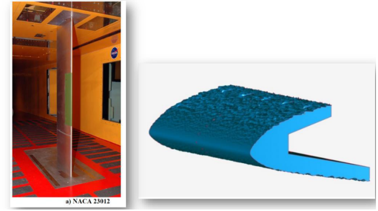
- **Growing recognition**: Ice accretion has gained increasing attention in recent years due to its critical role in the aircraft certification process.
- **Broad participation**: The workshops attract leading academic institutions and major industry players, including NASA, ONERA, JAXA, CIRA, Bombardier, Boeing, Polytechnique Montréal, and Politecnico Milano.
- **Increased complexity**: Ice shape prediction has evolved significantly, now encompassing both **2D** simulations and advanced **3D** configurations.
- **Multi-layer modeling**: The use of a multi-layer ice prediction approach is becoming essential, enabling more accurate modeling of ice accretion across multiple layers.
- **Parametric studies**: These studies investigate the effects of varying key parameters, such as temperature, which significantly increase computational demands.

HLPW edition	Number of simulations to perform	Max mesh size (#cells)	Number of participants
1st-2021	16	80M	20
2nd-2023	9	22M	15

Ice Prediction Workshops



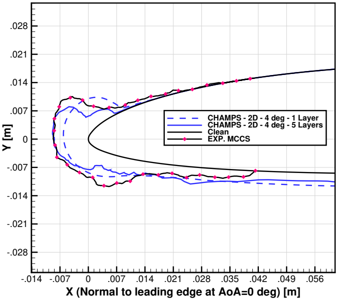
(a) Glaze Ice Conditions



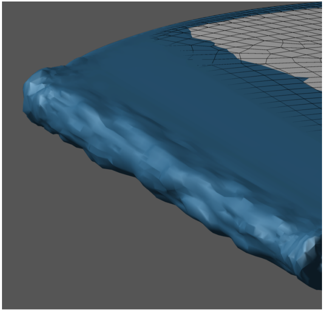
(b) Rime Ice Conditions

(c) 3D Hybrid Configurations

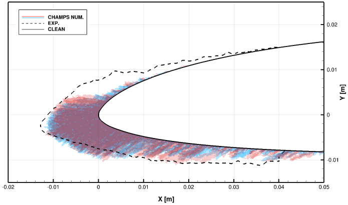
Ice Prediction Workshops



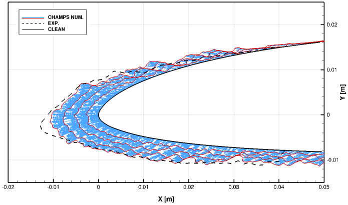
(a) 2D Ice Accretion



(b) 3D Ice Accretion



(c) Stochastic Ice Accretion



(d) Multi-layer Stochastic Ice Accretion

Canadian Public investments

Federal investments

On March 2023, Innovation, Science and Economic Development (ISED) Canada awarded funding of up to $228.3 million to The Digital Research Alliance of Canada for 2023-2025. This includes

- $120.6M for Infrastructure Renewal
- $8.2M for data centers upgrade

Provincial investments

This federal investment alongside provincial government investments will be used to renew the computing facilities of The Digital Research Alliance of Canada:

- (ON) $26.2M from the Alliance and $26.2M from the government of Ontario to the University of Toronto to upgrade the Niagara cluster.
- (ON) $21.8M from the Alliance and $21.2M from the government of Ontario to the University of Waterloo to upgrade the Graham cluster.
- (QC) $19.35M from the Alliance and $19.35M from the government of Quebec to McGill University to upgrade the Beluga cluster.
- (BC) $41.5M from the Alliance and $24.5 from the government of British Columbia to Simon Fraser University to upgrade the Cedar cluster.
- (BC) $10.3 from the Alliance and $6.1 from the government of British Columbia to the University of Victoria to upgrade the Arbutus cloud cluster.

Prof. Laurendeau 2025 Competition requests

Table 2: Summary Table of Compute Request Size

Project		Core years	GPU/VCPU years	Storage(Gb)
Aerodynamics	*Steady*	*60.0*	*0.0*	0.0
	LES	*2602.0*	*0.0*	0.0
	AI Database	*90.0*	*0.0*	0.0
Multi-physics	*Icing*	*800.0*	*0.0*	0.0
	Aero-elastic	*20.0*	*0.0*	0.0
Multi-user environment	Software	0.0	0.0	50.0
Totals:		**3572.0**	**0.0**	**50.0**

HPC costs

- 2023/2024 1113 core-year, $119 792 (actual)
- 2024/2025 3572 core-year, $384 453 (request)

Outlook for Aerospace HPC needs

Disciplines

- Aerodynamics (e.g. novel configurations)
- Propulsion (e.g. open prof-fan)
- Combustion (e.g. SAF, Contrails)
- Certification (1-pilot cockpit, Unmanned Aerial Vehicles)

Source: Internet

Outlook for Aerospace HPC needs

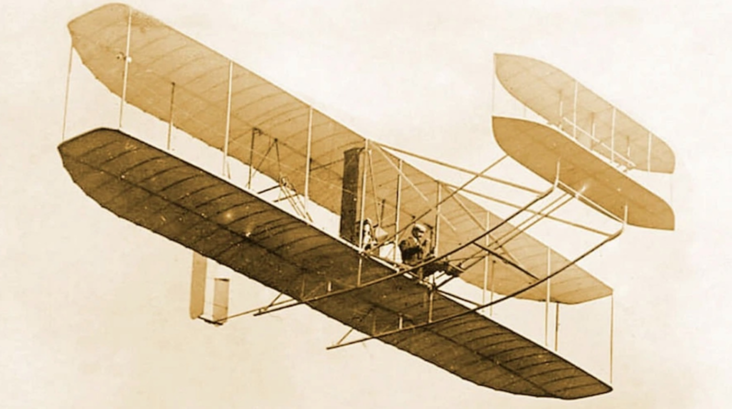
HPC hardware

- CPU
- GPU
- Hybrid
- Mixed-precision

HPC software

- efficient, parallel software, single-disciplinary, analysis
- efficient, parallel software, multi-disciplinary, analysis
- efficient, parallel heterogeneous hardware, analysis design
- Possibilities offered by Chapel and similar languages are pointing to right direction

Questions? Comments?



Wright brothers, Wikipedia