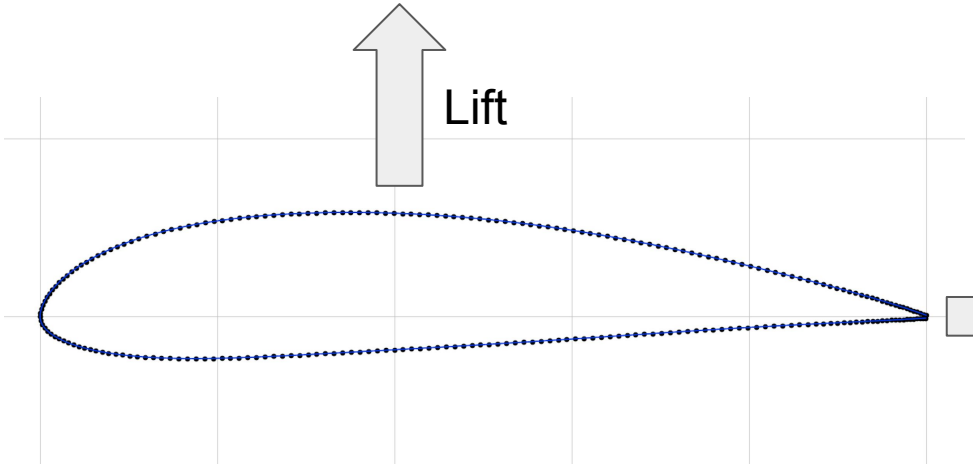
Big Compute Clash 2020

Problem Statement

(in the clouds)

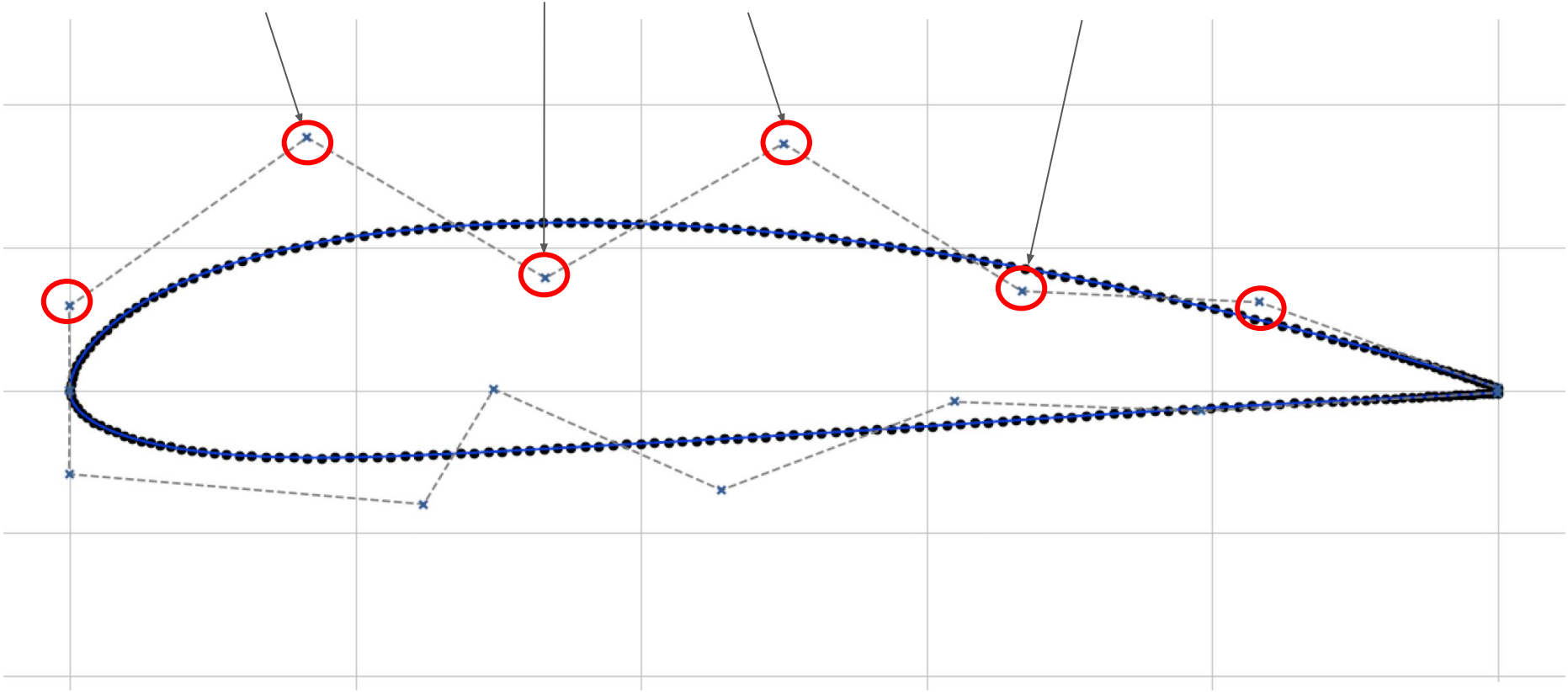
Primary Objective: Maximize Lift/Drag (L/D) of an airfoil



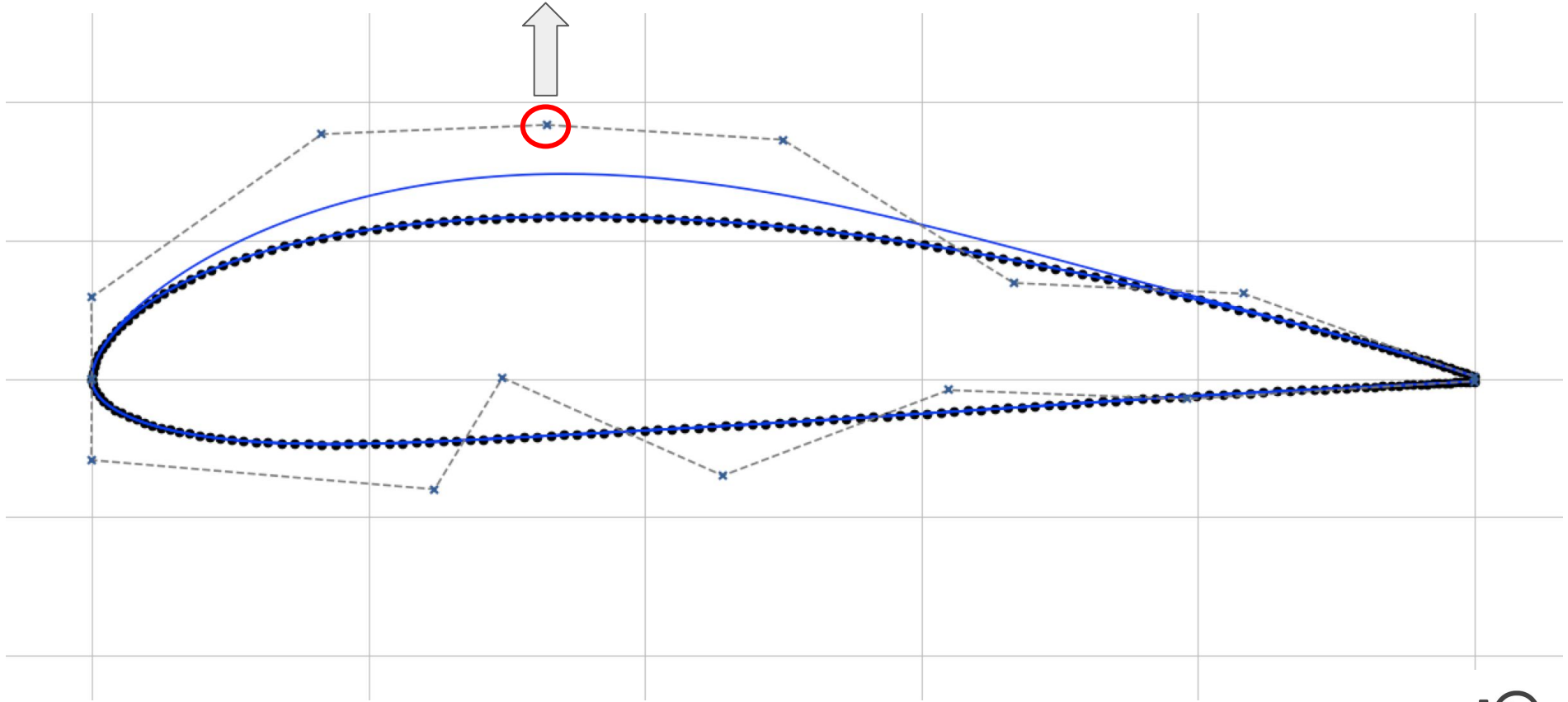
Want to drive up the value of lift while keeping drag minimal

- Control the shape of the airfoil by perturbing Bezier control points
- Fix angle of attack, Mach, and other flow characteristics
- Given a finite compute budget

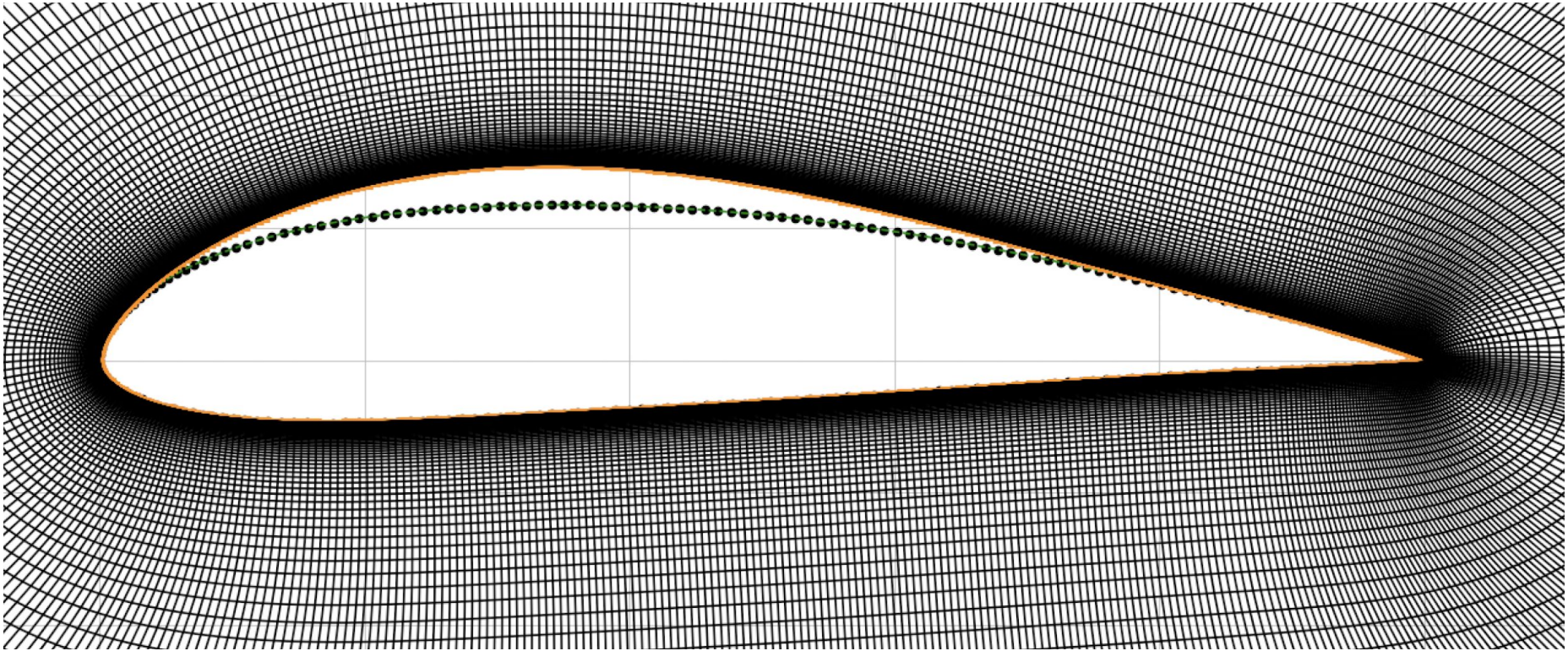
Control Points define a spline for each surface of the airfoil



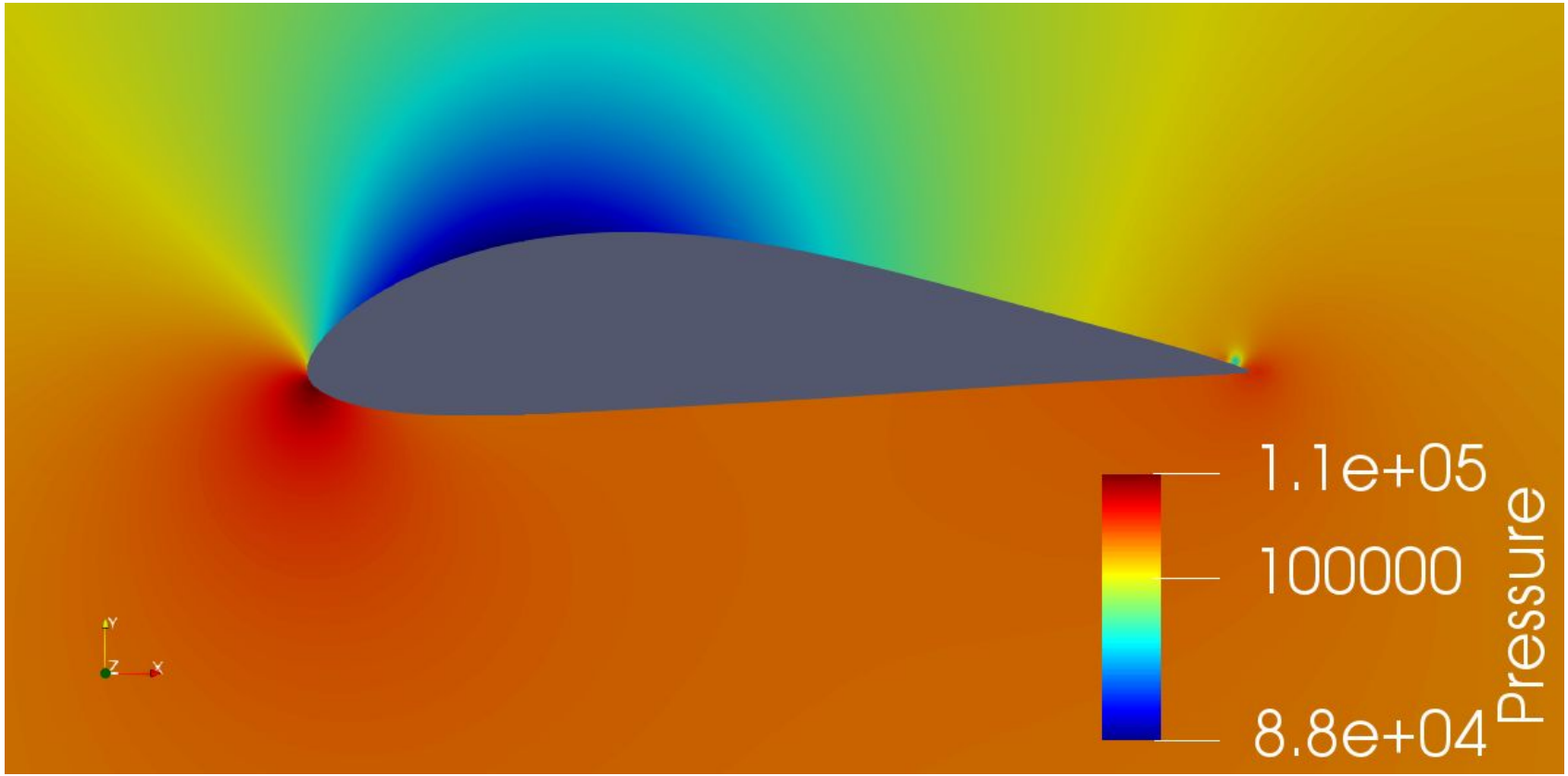
Moving control points will create a new curve for the airfoil surface



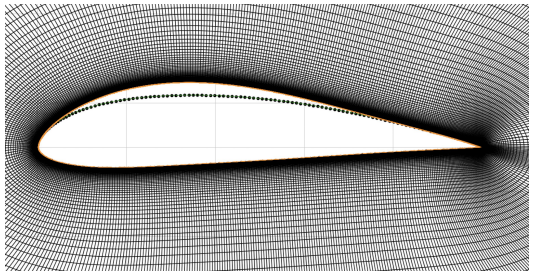
Script will then generate a computational mesh for new surface



Computational mesh run through SU2 CFD solver



#	X	Y
1	[-1,1]	[-1,1]
2	[-1,1]	[-1,1]
3	[-1,1]	[-1,1]
...		



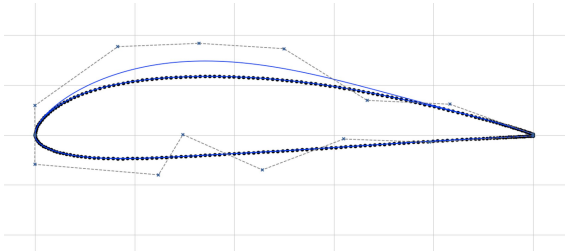
Input Perturbations

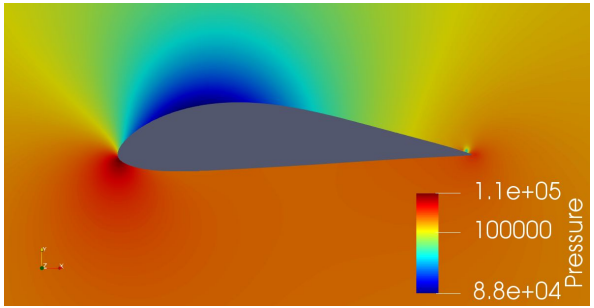
Control Points Perturbed and New Airfoil Surface Generated

Mesh Generated for Surface (course, fine)

Mesh Run with CFD Solver SU2

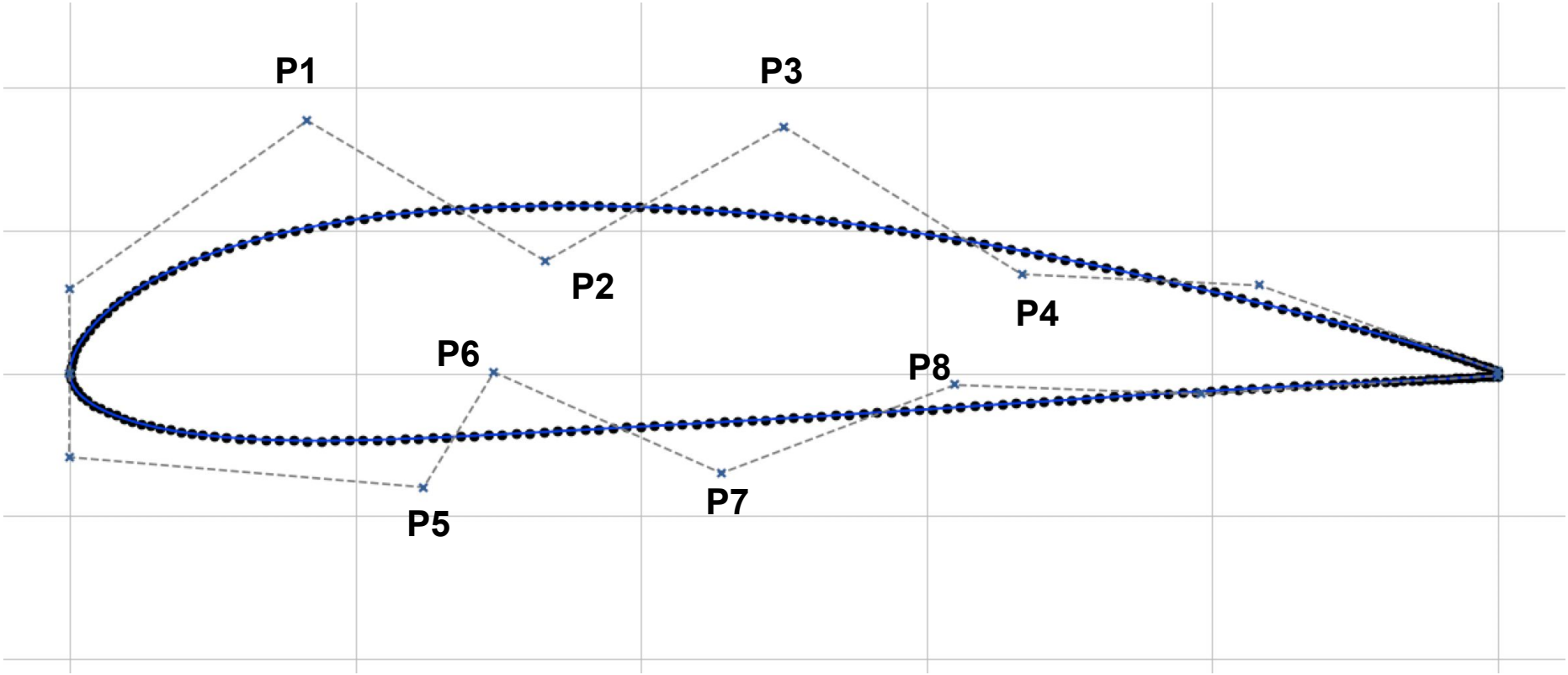
Performance Metric L/D





[BIG] COMPUTE

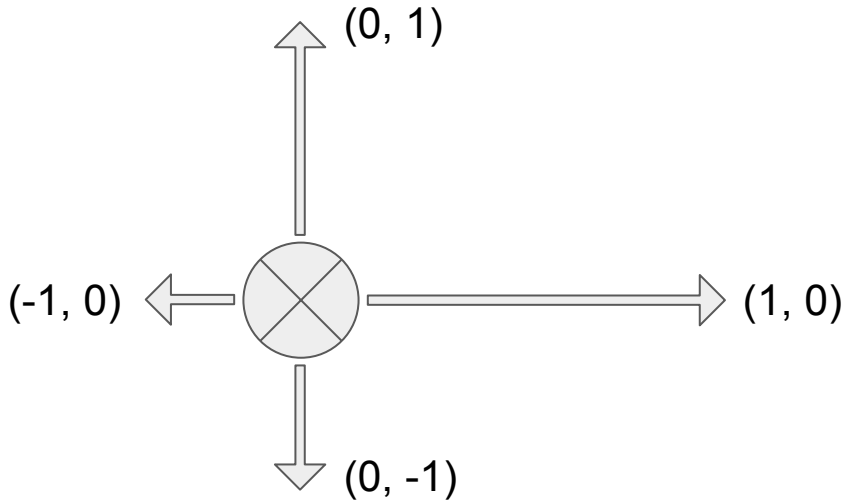
Labeling of points for input



User will provide an input file the maps to a perturbation of a control point - these are normalized onto [-1, 1] with preset bounds

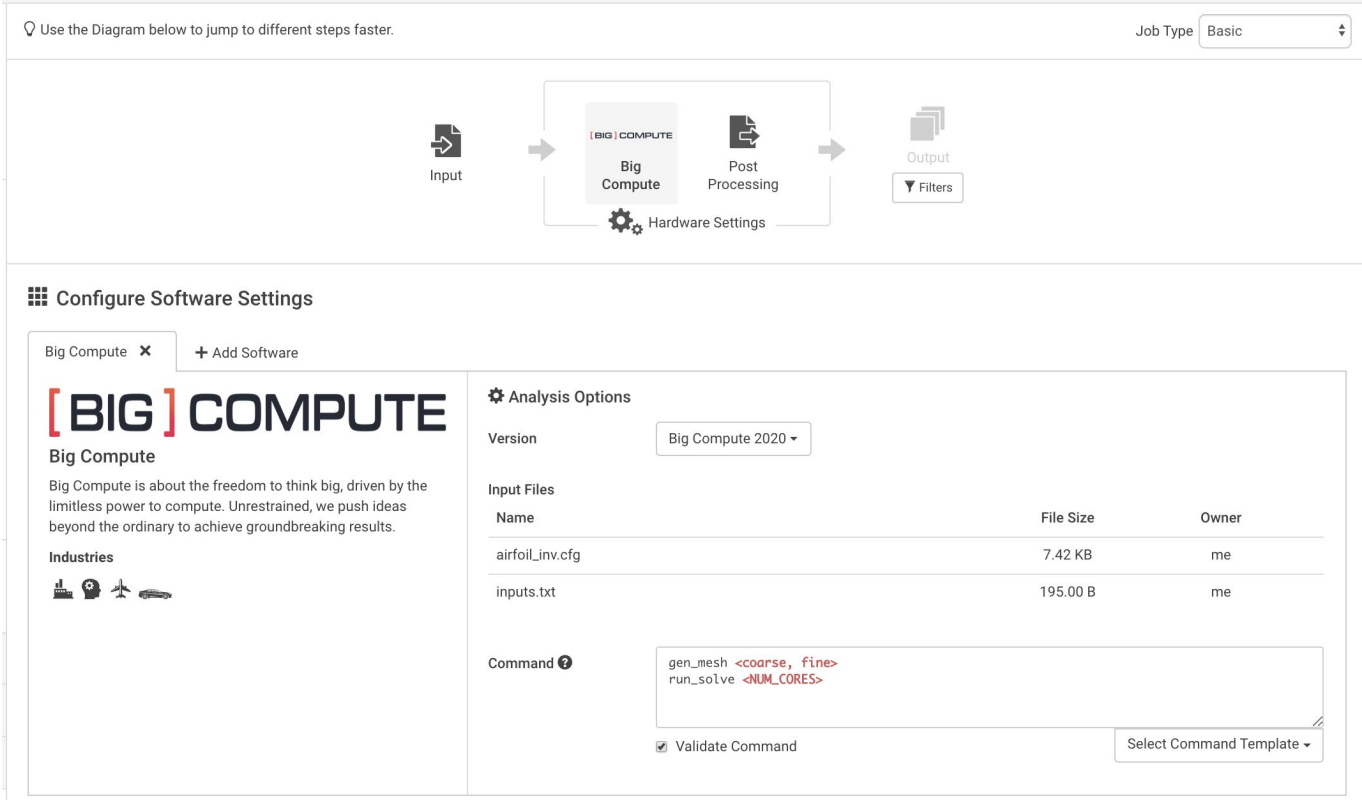
inputs.txt

Point #	X-coord	Y-coord
1	0	0.50
2	-0.10	-0.25
3	0.40	-0.50
4	-0.30	-0.25
5	0.50	-0.25
…		
...		



* Pattern shown for demonstration only, not all (if any) points will map to this pattern

rescale



Need two inputs:

- inputs.txt
- airfoil_inv.cfg

Run two commands:

- `gen_mesh` (with arg either coarse, medium, fine)
- `run_solve` (with arg the number of cores to run on)

Outputs:

- Some pretty pictures
- Solution files
- output.txt summary

Example Job

https://platform.rescale.com/jobs/fqvWo/setup/input-files/

Other controls

- Can use either a ***coarse***, ***medium***, or ***fine*** mesh - coarse will run faster but might not arrive at acceptable solution

- Control the solver iterations (**EXT_ITER**) in the configuration file **airfoil_inv.cfg**

  …
  % Number of total iterations
  EXT_ITER=500
  …

  fewer iterations will run faster, but the solution may not pass

- Run on more cores - this will generate the solution faster, but will cost slightly more (parallel efficiency). You have a **$25 limit**, use it wisely.

The Catch: Solution Must be Converged!

- All non-analytic numerical solutions carry some type of error
- This error must be below a threshold
  - This will be checked automatically and printed to an output summary

How do I fix a non-converged solution?

- Increase solver iterations
- Restart solver (subtle numerical trick)
- Change mesh quality
- Throw away geometry

output.txt

Converged: Yes/No

Lift/Drag: 80

Lift: 1.1

Drag: 0.1375

Success Metrics: Maximize Lift/Drag of an airfoil

- Quantitative value of L/D. Final L/D must be run on fine mesh.
- Approach. Explore the design space as efficiently as possible. Bonus points for creativity.
- 3 finalists will be selected and will present their solution

$$\mathbf{SCORE} = f(\mathrm{L/D}) + f(\mathbf{approach}) + f(\mathbf{presentation})$$

/10 **/20** **/5**

Winner takes home the goods!

[BIG] COMPUTE

Final Submission

- Rescale job with your best L/D
  (Make sure the solution has converged with a fine mesh)

- Presentation, 5 slides or less
  (Google slides, make sure it's publicly accessible)

More details in instruction email.

Questions?

Tips and Tricks

- Interactively work on the problem by inserting a **sleep inf** in the command on the Rescale UI

- You can drop the **EXT_ITER** down to only a few iterations to get quick look at the mesh generated (want to be running interactively)

- You can restart solver from a previous solution by changing the flag (**RESTART_SOL**) in the configuration file **airfoil_inv.cfg**

  ...
  % Restart solution (NO, YES)
  RESTART_SOL= NO

  ** the file indicated by the field **SOLUTION_FLOW_FILENAME** must in the directory the code is run in, and from the same mesh