MATTHEW FISCHELS

Aerospace Engineering Department

B.S. Aerospace Engineering, Iowa State University

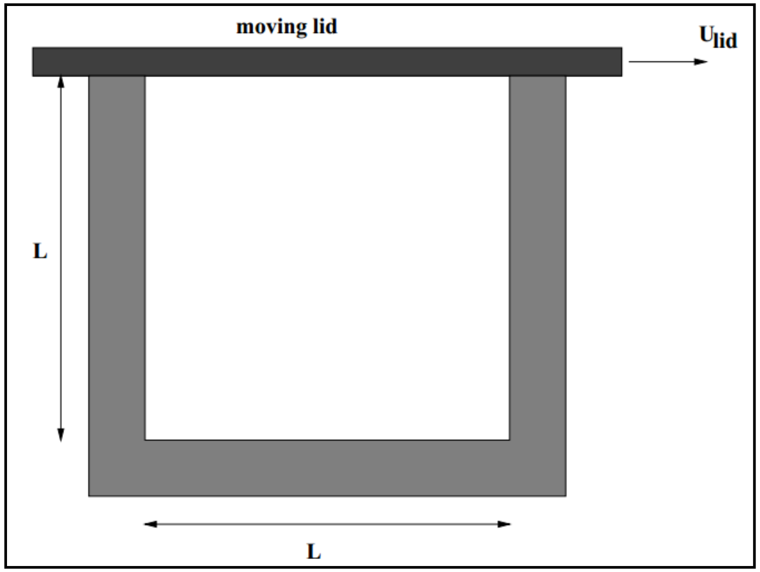
Major Professor : Dr. R. Ganesh Rajagopalan

Area of Research

- Aerodynamics
- Computational Fluid Dynamics (CFD)

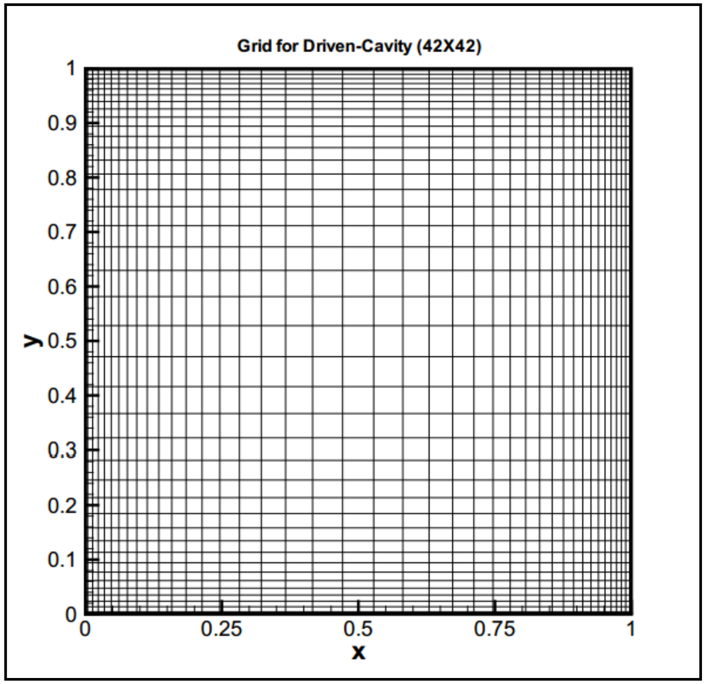
What is CFD?

- Use of numerical methods to solve fluid flow problems
  1) Domain – B.C.



What is CFD?

- Use of numerical methods to solve fluid flow problems
  1) Domain – B.C.
  2) Descritize – Grid/Mesh

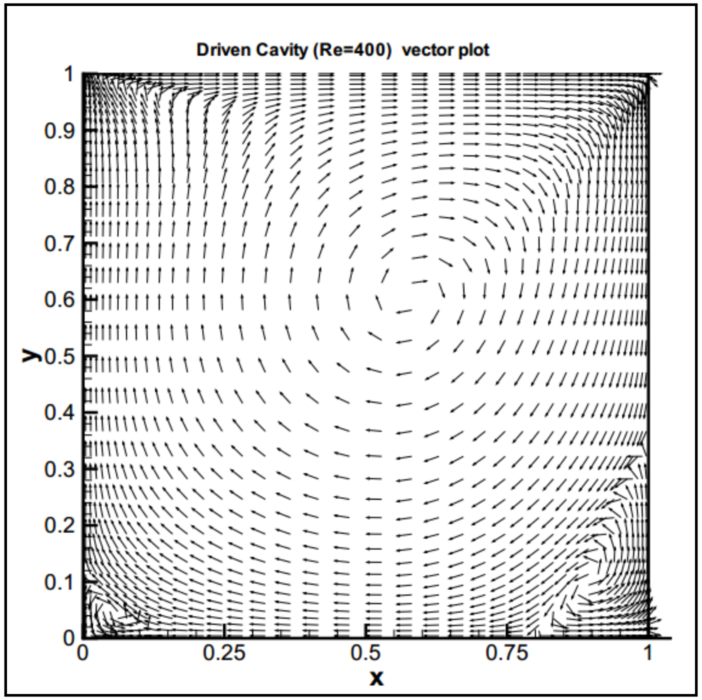


What is CFD?

- Use of numerical methods to solve fluid flow problems
  1) Domain – B.C.
  2) Descritize – Grid/Mesh
  3) Solve Eqs. (N-S, Euler, …)

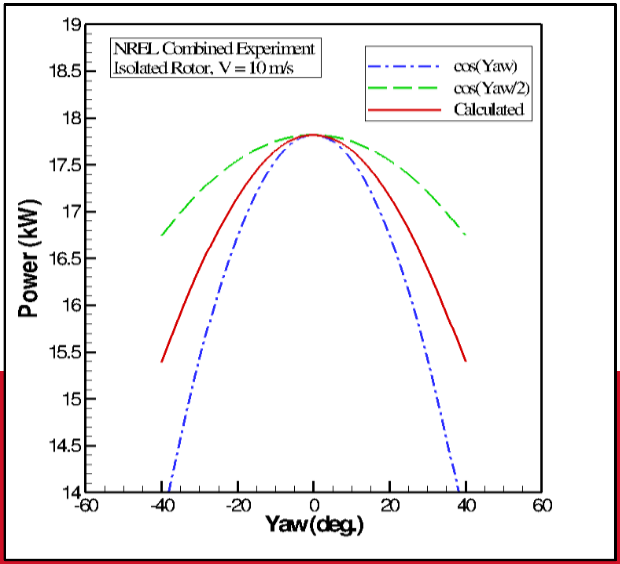
     with certain assumptions

     (incompressible, inviscid)



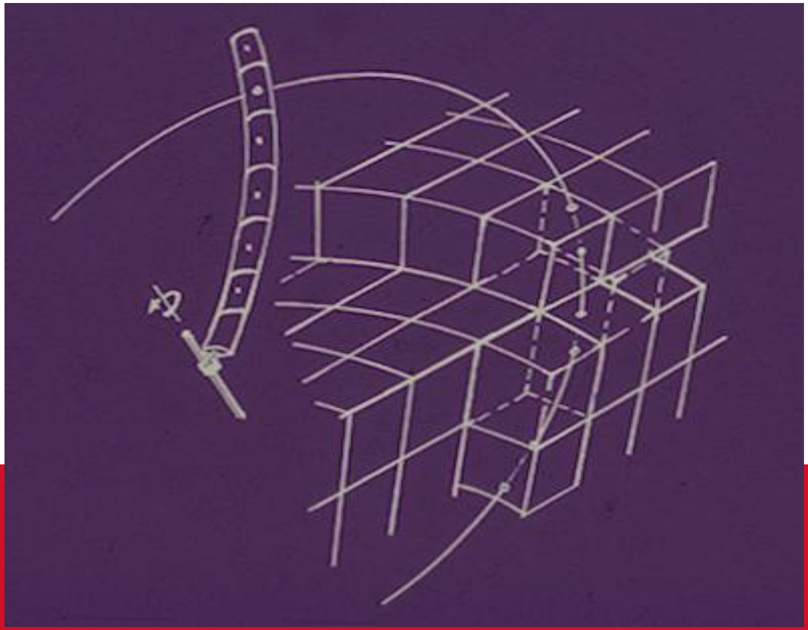
CFD in Wind Energy

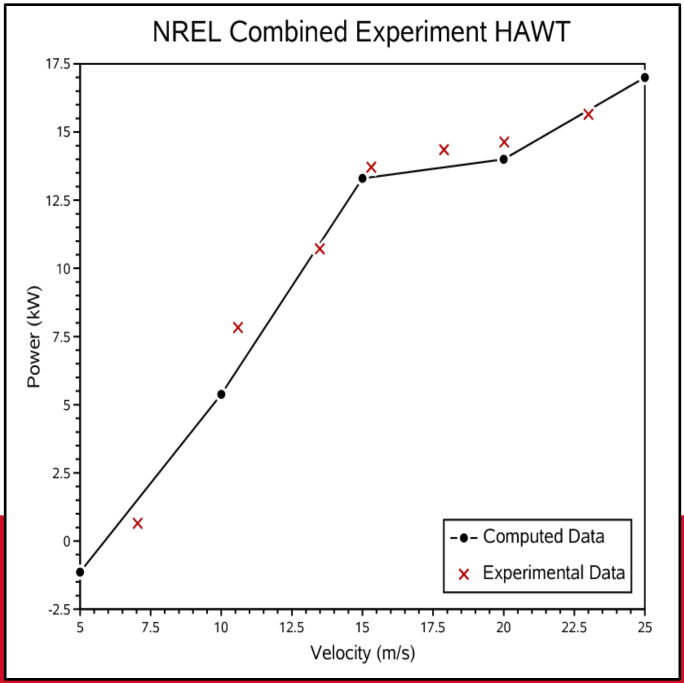
- Design
  - Single Turbine, Wind Farm
- Analysis
  - Yaw Effects, Wake Interference



Blade Modeling

- Momentum Source Modeling of Blades
  - Coarser grids than modeling blade geometry
  - Save time without losing much accuracy

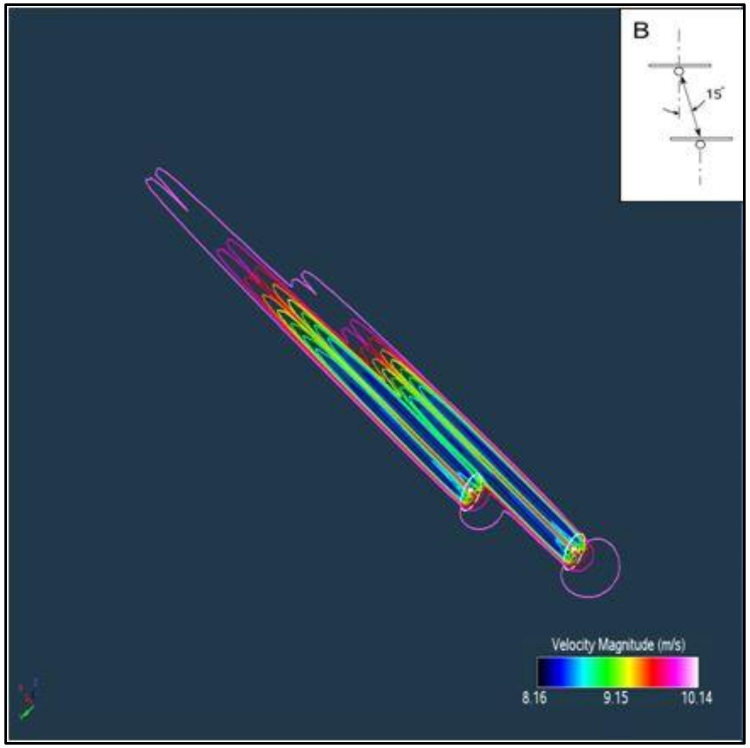




Problem

- Computational tools for analyzing wind turbine aerodynamics need to be improved
- Primary interest is decreasing the run time of CFD

Problem



- Example: Two HAWTs interference study
- ~3 days per run for steady solution
- Reduction in run time would increase design and analysis capability

		wall time required per revolution [days]				
# of processors	memory/ node [MB]	Lion-xl	Mufasa	NREL cluster Std	NCSA cluster	NASA cluster
16	297	14	42	15	10	11
32	192	6	12	10	5	5.4
64	140	-	8	-	2.4	3.5
128	113	-	5	-	1.7	1.8

Solutions

- Implicit Methods
- Using potential flow solutions as a starting solution
- Multi-Grid
- Graphics Acceleration (GPU)

Implicit Methods

- Navier-Stokes Equations – P.D.E.

Momentum Conservation

$$\frac{\partial(\rho\vec{u})}{\partial t} + \rho\vec{u}\cdot\nabla\vec{u} = -\nabla P + \nabla\cdot(\mu\nabla\vec{u}) + S_u$$

Mass Conservation

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\vec{u}) = 0$$

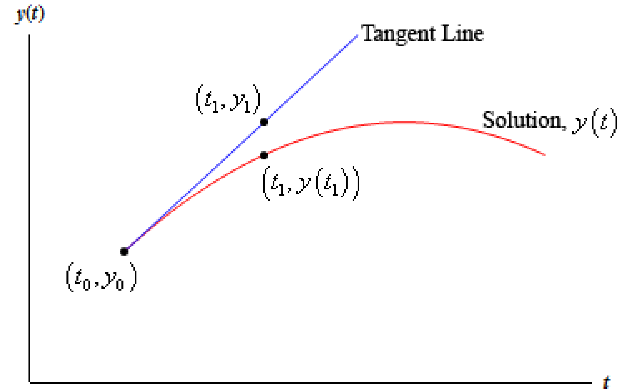
- Spatially discretized – O.D.E. (non-linear)

$$\frac{\partial u}{\partial t} = F\big(u(t)\big)$$

Implicit Methods

- Explicit

$$\frac{\partial u}{\partial t} = F(u(t)) \quad => \quad \frac{u^{n+1} - u^n}{\Delta t} = F(u^n)$$

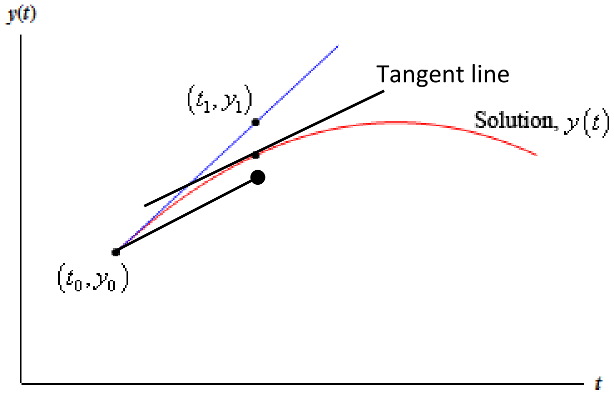


  - Advantage: RHS is know, simply march in time
  - Disadvantage: Small times steps are needed

- <u>Less work per step – More steps</u>

Implicit Methods

- Fully Implicit

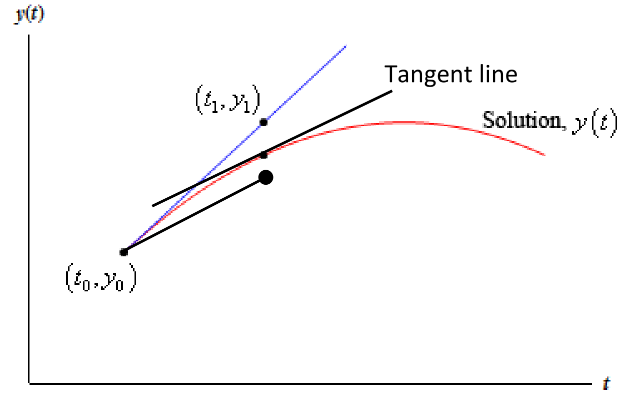


$$\frac{\partial u}{\partial t} = F\big(u(t)\big) \quad => \quad \frac{u^{n+1} - u^{n}}{\Delta t} = F(u^{n+1})$$

  – Advantage: More stable for large time steps
  – Disadvantage: RHS is function of the next time step – need to solve a system of equations

- <u>Fewer Steps – More work per step</u>

Implicit Methods



- Crank-Nicolson (trap. rule)

$$\frac{\partial u}{\partial t} = F(u(t)) \quad => \quad \frac{u^{n+1} - u^n}{\Delta t} = \frac{1}{2}[F(u^{n+1}) + F(u^n)]$$

  – Same amount of work as implicit
  – More accurate in time

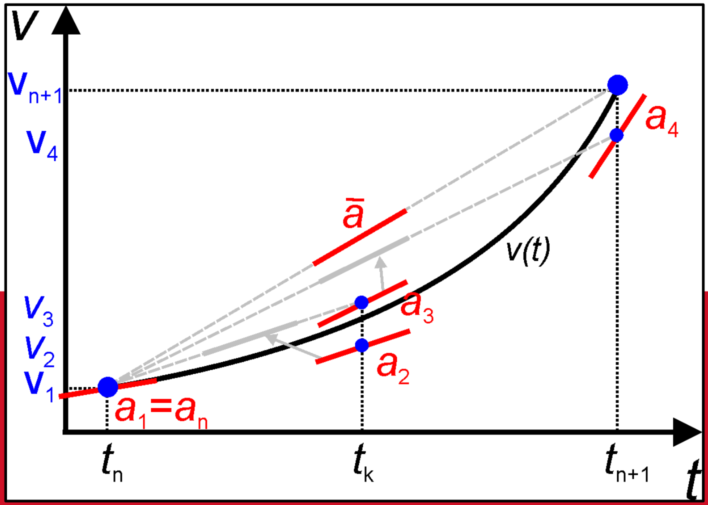
- Can use any weighting of F's

$$\frac{u^{n+1} - u^n}{\Delta t} = [\alpha F(u^{n+1}) + (1 - \alpha)F(u^n)]$$

These three methods only work for very small time steps or with iterations within the time step (non-linear)

Implicit Methods

- Explicit Runge-Kutta Methods
  - Multi-step explicit
  - Averages different slopes
  - More work than simple explicit but more stable
  - Does not require iterations for small time steps



Implicit Methods

- Implicit Runge-Kutta Methods
  - Multi-step implicit
  - Averages different slopes
  - Each stage has a system of equations to solve, more work per step
- Not as common, where research can be done

Implicit Methods

- Iterative Implicit/C-N
  - Steady Problems – large time step
- Explicit Runge-Kutta
  - Unsteady Problems – less work per step
- Implicit Runge-Kutta – Research

Questions?