APISAT 2019

2019 ASIA PACIFIC INTERNATIONAL SYMPOSIUM ON AEROSPACE TECHNOLOGY

SURFERS PARADISE MARRIOTT RESORT, GOLD COAST
4 – 6 DECEMBER 2019

ENGINEERS AUSTRALIA

Aerospace Systems Design III
Wednesday, Dec 4, 2019
4:10 PM - 6:10 PM
Hinterland Room 2

Propeller Design Framework using Open Source Tools

Daniel de Vries, Wanbo Liu, **Willem A. Anemaat**
Design, Analysis and Research Corporation

APISAT 2019

ENGINEERS AUSTRALIA

ROYAL AERONAUTICAL SOCIETY
AUSTRALIAN DIVISION

DARcorporation

- Formed in 1991 by Dr. Jan Roskam and Dr. Willem Anemaat
- Located in Lawrence, Kansas
- Mission:

Provide Integrated Aircraft Design, Development and Consulting Services

  - Airplane Analysis and Design Consulting Services
  - Market, Support and Develop Airplane Design and Analysis Software
  - Market and Distribute Airplane Design Books written by Dr. Jan Roskam
  - Wind Turbine Design

Preliminary Design Propeller Analysis Tools

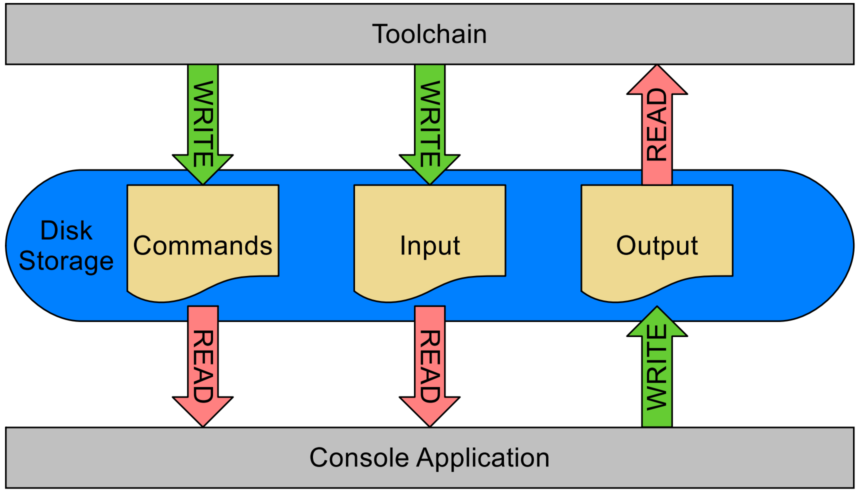
- 2D Airfoil Analysis: XFOIL
- Blade Element Method: XROTOR
- Coupled Analysis:
  - Local Conditions at Airfoils Fed Back Into XFOIL

Original XFOIL/XROTOR Code

- XFOIL 6.99 & XROTOR 7.55 Written in FORTRAN 77
- “Black-box” Console Applications Without GUI
- Interaction Only Via Keyboard or Files on Hard Disk
- 3^{rd} Party Frameworks Must Rely on Disk I/O for Integration

Integration of Console Applications within Toolchains

- 6 Disk I/O Operations Per Run Case
- Full Memory Allocation Each Run
- Cannot Debug What Happens inside Console Application
- Difficult to Monitor Computational Progress



Inefficient & Ineffective

Modernization of XFOIL/XROTOR Code (approach)

- Stripped Down Code
  - Retaining Only Essential Analysis Routines
- Upgrade from Old FORTRAN 77 to Modern Fortran 2018
- Compiled as Shared Library with Condensed API
- Python Package Exposing Functions in Object-Oriented Way

Modernization of XFOIL/XROTOR Code (results)

- New Libraries Give Identical Results:
  - Functionality Retained
- No Disk I/O and Reduced Memory Footprint:
  - Code is up to 48% Faster
- Direct Access to Functionality from High-Level Language:
  - Improved Ease of Use

Propeller Model

- Discretization:
  - Class-Shape Transformation (CST)
  - Reduction of Nr. Of Design Variables

$$z(x) = \left[C_{N_2}^{N_1}(x) \sum_{\mu=1}^{n} A_\mu S_\mu^n(x) \right] + (1-x)\cdot \delta z_{\text{start}} + x \cdot \delta z_{\text{end}}$$

$$C_{N_2}^{N_1}(x) = x^{N_1} \cdot (1-x)^{N_2}$$

$$S_\mu^n(x) = K_\mu^n x^\mu (1-x)^{n-\mu}$$

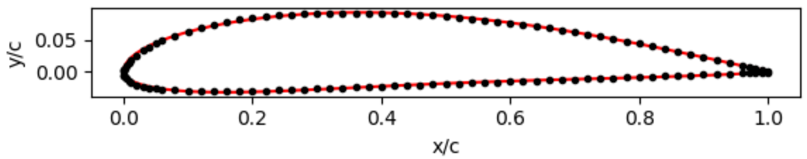
$$K_\mu^n \equiv \binom{n}{\mu} = \frac{n!}{\mu!(n-\mu)!}$$

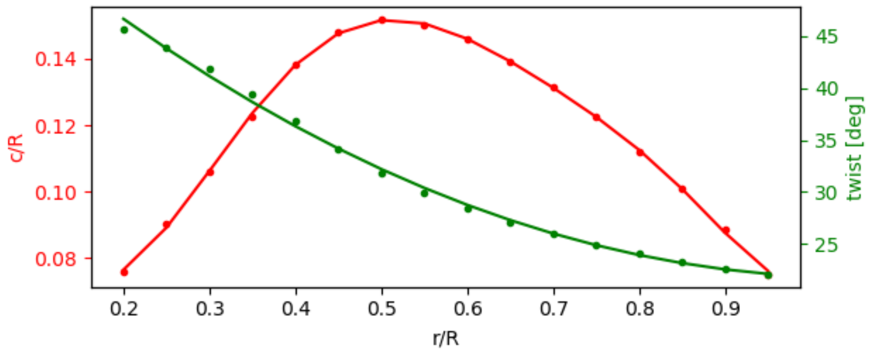
CST

Description	Continuous	Discrete
Hub radius	r_{hub}	r_{hub}
Tip radius	r_{tip}	r_{tip}
Chord distribution	$c(r)$	$A_{c,j}$
Twist distribution	$\beta(r)$	$A_{\beta,k}$, $\delta\beta_{\text{start}}$, $\delta\beta_{\text{end}}$
Blade cross section upper coordinates	$\left[r_i,\ z_{U,i}(x/c)\right]$	$\left[r, A_{U,l}, (\delta z_{\text{end}})_U\right]_i$
Blade cross section lower coordinates	$\left[r_i,\ z_{L,i}(x/c)\right]$	$\left[r, A_{L,l}, (\delta z_{\text{end}})_L\right]_i$
Number of blades per propeller	n_{blades}	n_{blades}

Propeller Model (cont.)

- Just 3 Coefficients Per Curve Sufficient for Good Propeller Fit





- 20 Design Variables

Coupling XFOIL to XROTOR

- XFOIL Gives Aerodynamic Polars as Tables
- XROTOR Needs Aerodynamic Data as its Own Polar Model
- Translation is Needed from Table to Model
- Least-Squares Fit of Model to Data is Obvious Approach

Coupling XFOIL to XROTOR (cont.)

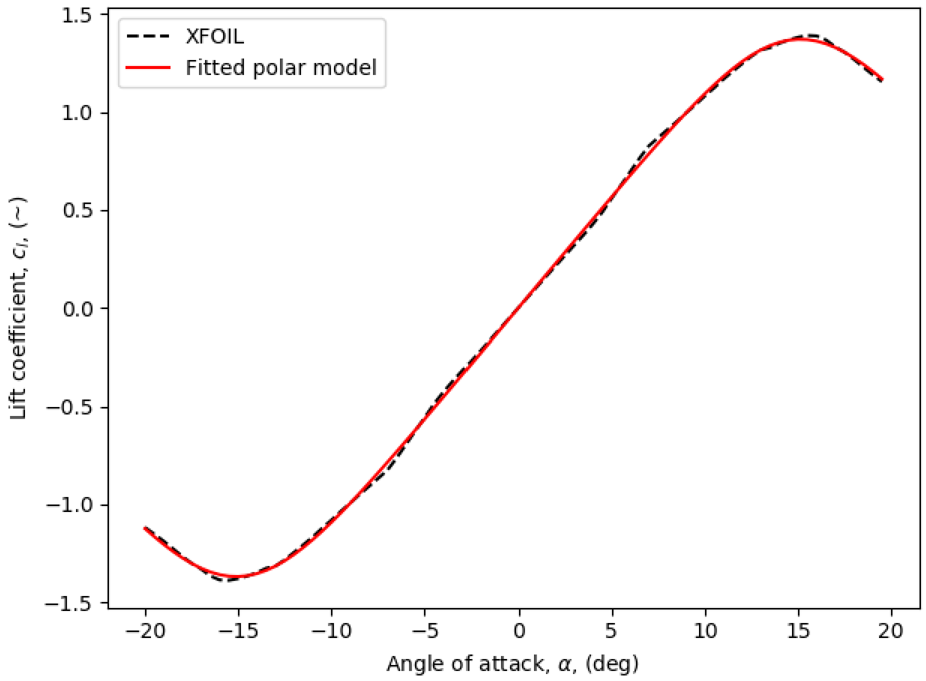
- XROTOR Polar model:

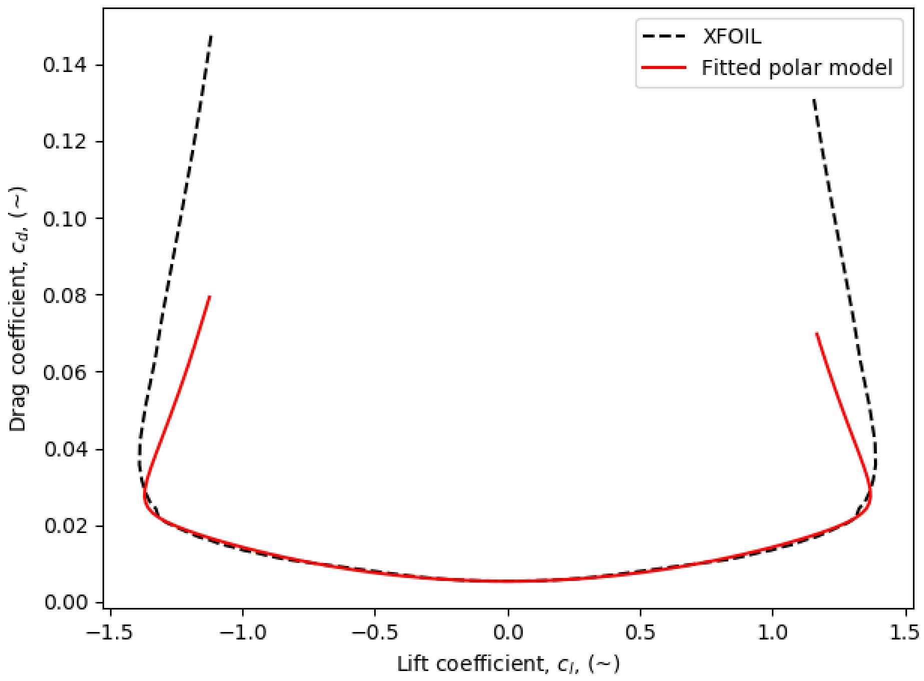
Description	Symbol	Units
Angle of attack at zero lift	α_0	deg
Maximum lift coefficient	$c_{l_{\max}}$	~
Minimum lift coefficient	$c_{l_{\min}}$	~
Linear lift curve slope	c_{l_α}	1/rad
Post stall lift curve slope	$c_{l_{\alpha_{\mathrm{stall}}}}$	1/rad
Lift coefficient offset from stall onset to full stall	$\delta c_{l_{\mathrm{stall}}}$	~
Minimum drag coefficient	$c_{d_{\min}}$	~
Lift coefficient at minimum drag coefficient	$c_{l_{c_{d_{\min}}}}$	~
Quadratic drag polar coefficient	$\frac{\partial c_d}{\partial {c_l}^2}$	~
Pitching moment coefficient	c_m	~
Critical Mach number	M_{crit}	~

Intuitive characteristics of lift curves and drag polars

Coupling XFOIL to XROTOR (cont.)

- Works Well for Simple Polars:





Coupling XFOIL to XROTOR (cont.)

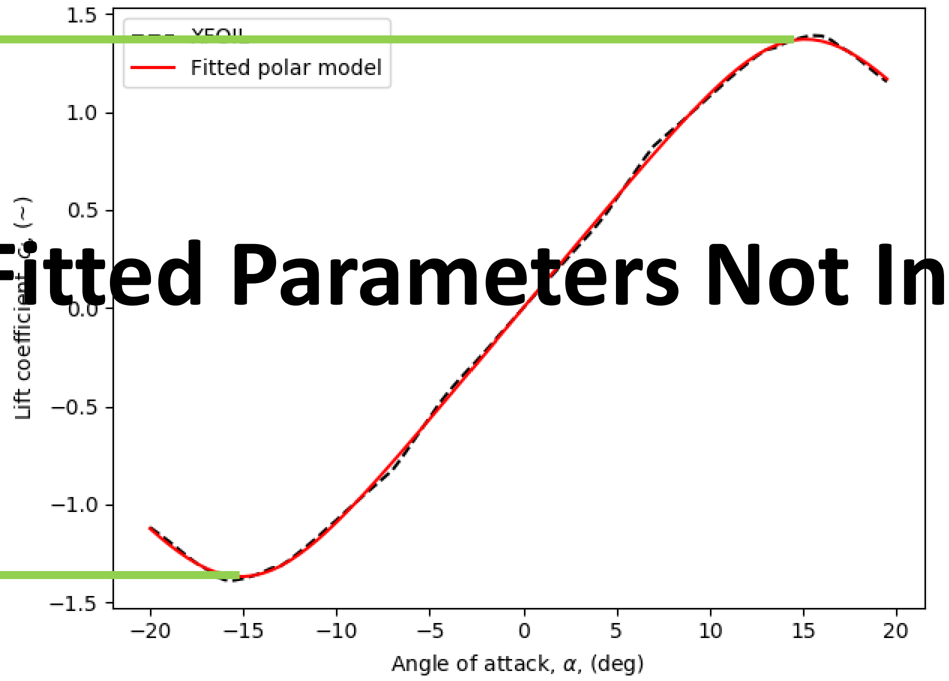
- However, Resulting Parameters for this Fit Are:

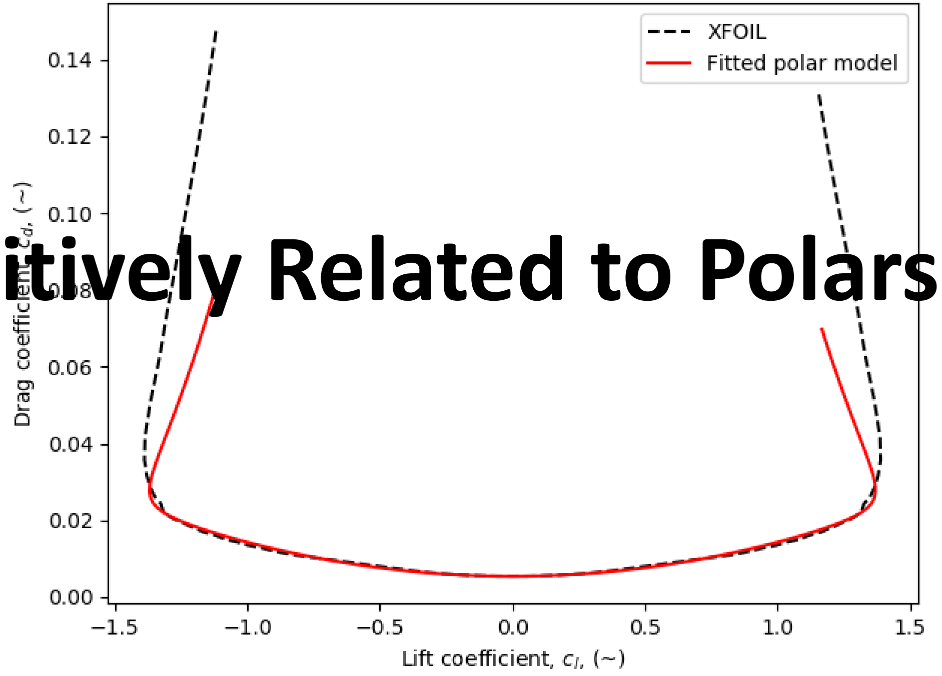
$$\alpha_0 = 0.00\,\text{deg}\,,\ c_{l_{\max}} = 1.84\,,\ c_{l_{\min}} = -1.84\,,\ c_{l_\alpha} = 6.60\,\text{rad}^{-1}\,,\ c_{l_{\alpha_{\text{stall}}}} = -8.71\,\text{rad}^{-1}\,,\ \delta c_{l_{\text{stall}}} = 0.29\,,$$

$$c_{d_{\min}} = 5.37\times10^{-3}\,,\ \frac{\partial c_d}{\partial {c_l}^2} = 8.76\times10^{-3}\,,\ c_m = 2.49\times10^{-4}\,,\ M_{\text{crit}} = 0.81\,.$$

$c_{l_{\max}} \neq 1.84$

$c_{l_{\min}} \neq -1.84$

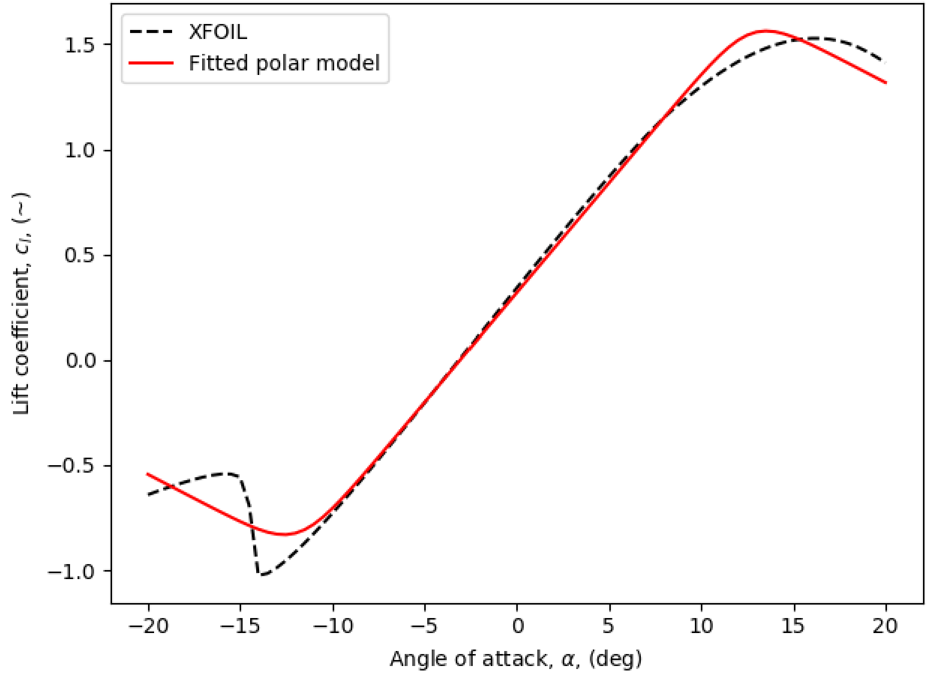


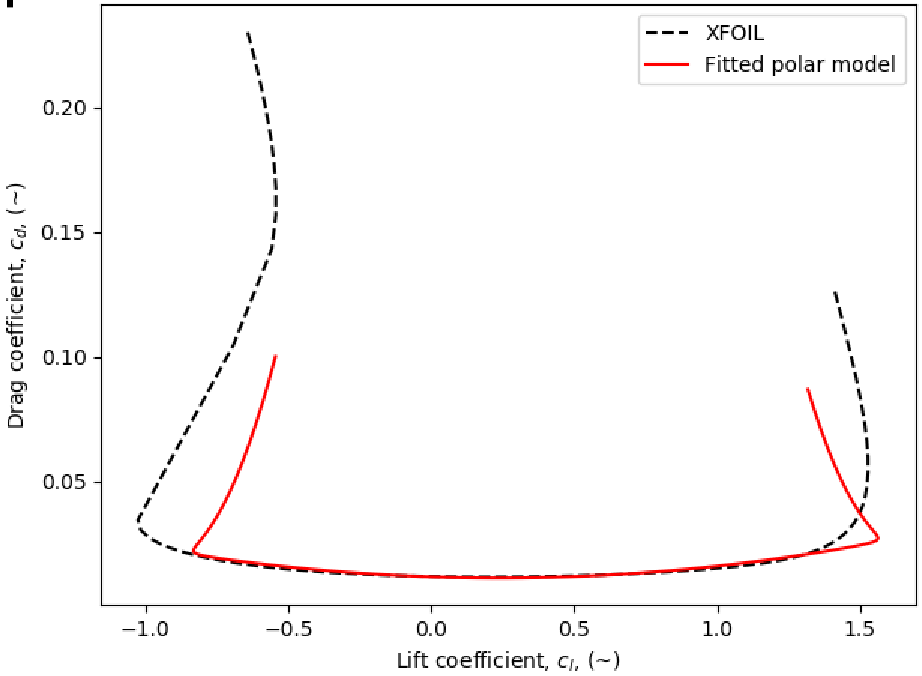


Fitted Parameters Not Intuitively Related to Polars!

Coupling XFOIL to XROTOR (cont.)

- Causes Problems for More Complex Polars:

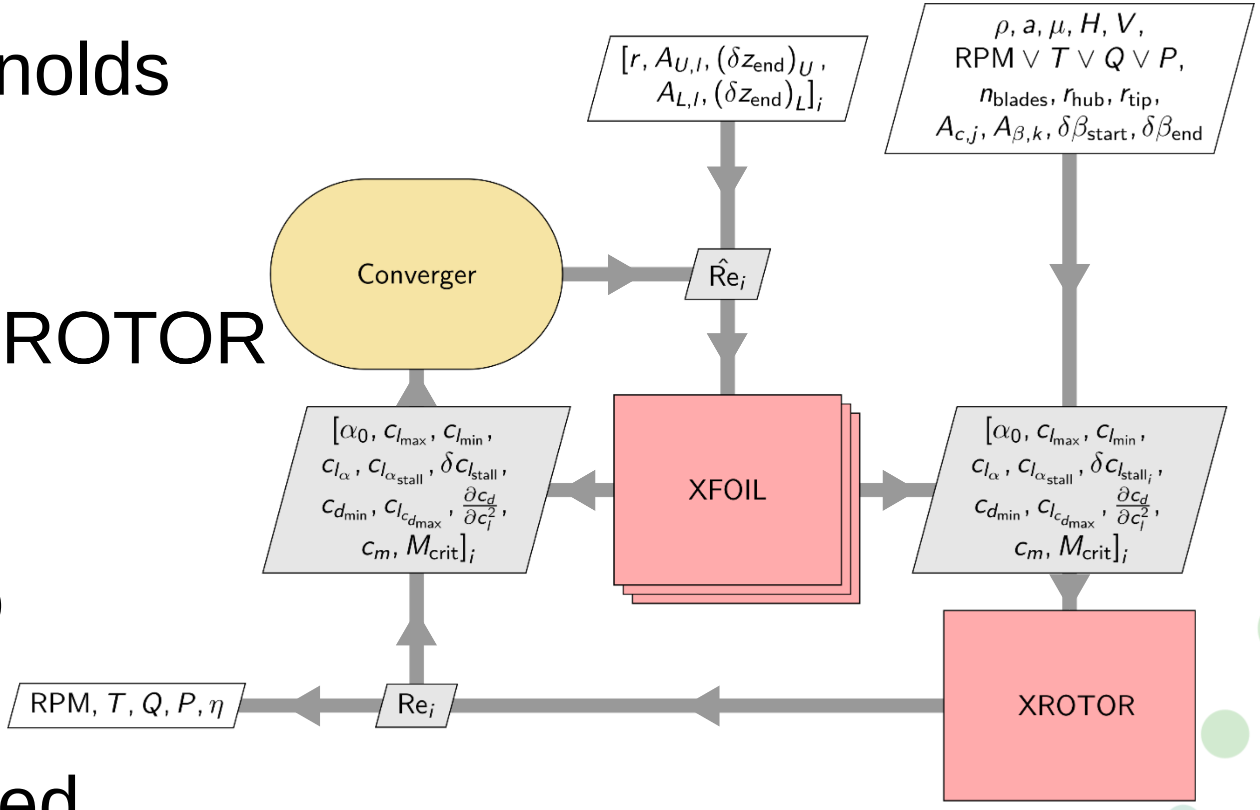




- This is Core Difficulty in Coupling XFOIL to XROTOR

Coupling XFOIL to XROTOR (cont.)

- Coupled Analysis using Reynolds Numbers as Feedback
- XFOIL Polars are Fitted to XROTOR Model as Good as Possible
- Implemented in OpenMDAO
- Gauss-Seidel Converger Used

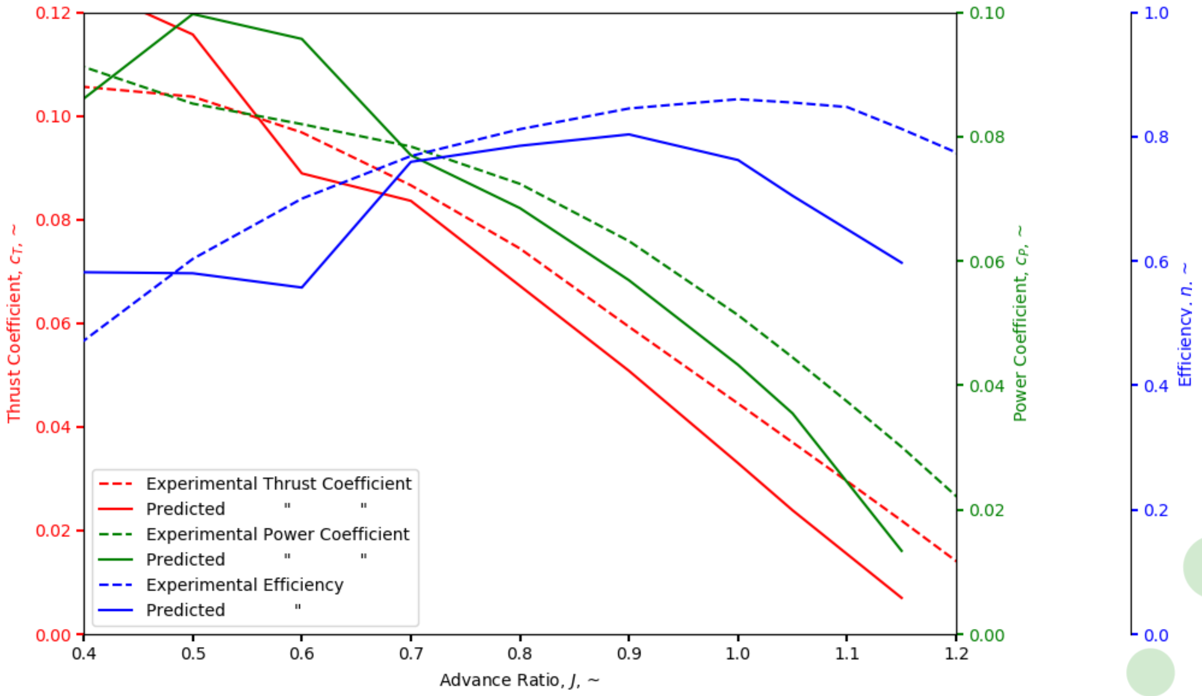


Results

- Reference Propeller:
  - 2-bladed 5868-9 (NACA Report No. 650)
  - Constant Clark-y Airfoil

- Resulting Perf. Curves Follow Roughly Same Trends

- Discrepancy is Due To Volatility of XROTOR Polar Model



Conclusions

- Modernizing Old Fortran Code Has Big Impact on Ease of Use and Speed
- Translation of XFOIL Polars to BEM Polar Model is Crux of the Problem
- If Polar Model Does Not Handle Every Polar Well Framework is Dead in the Water

Discussion

- Powerful Computing Resources Becoming More Accessible
- Allows Shift to High-Fidelity in Early Design Phases
- Modern Framework Could Exploit State-of-the-Art Continuous Adjoint RANS Solvers as Main Analysis Tool
- Parallelization is Key to Make this a Viable Solution

Discussion (cont.)

- Proposed Framework:
  - Continuous Adjoint RANS Solver from SU2
    - Proven, Powerful, Open-Source
  - Free-Form Deformation (FFD) Discretization
    - Lot of Design Freedom, Easy to Implement
  - OpenMDAO
    - Easy Control of Optimization
    - Easy to Integrate Extra Disciplines
  - Message Passing Interface (MPI)
    - Provides Scalability to High Performance Computing (HPC) Environments