Modelling Propellers in FINE/Open using OpenLabs

Matthew Anderson, KC Wong, and Patrick Hendrick

Abstract: With the prolific rise of unmanned aerial vehices (UAVs), interest is increasingly growing in optimising airframes to improve flight characterisitcs. Tools such as computational fluid dynamics (CFD) can be used to optimise the shape of UAV frames by giving a detailed understanding of how the flow interacts with the vehicle. The effects of the frame on the propellers (and vice versa) can be a dominant flow feature, especially for multirotors, however fully modelling propellers in CFD is a very involved task requiring intensive computational power.

Actuator and blade element disks provide a simple way of modelling the effects of a propeller in CFD by artificially applying a force into the domain, emulating the time-averaged effect of the propeller to reduce the computational requirements.

This paper presents an actuator disk model with three different loading profiles and a blade element disk model with two methods of calculating the thrust and the torque of the propeller. These models are implemented in FINE/Open using OpenLabs. The effect of using the differing models is shown, with results matching well with other implementations presented in literature.

1. Introduction

The biggest growth in commercial use of unmanned aerial vehicles has been through the use of multirotor aircraft for applications ranging from sport videography to environmental monitoring. Systems on these vehicles have in the past been the primary focus for development, but as they increasingly mature, airframes are becoming a major concern for improvement [1].

Fully modelling propellers in computational fluid dynamics (CFD) is a challenging endeavour requiring time dependant solutions, rotating domains and very complicated meshes. This can make modelling of aircraft where the propeller dynamics are important (such as multirotors) in CFD almost computationally prohibitive, and thus these airframes cannot be optimised to the same degree as other, more traditional platforms.

Actuator disk and blade element models take much of this complexity out of the problem by modelling the propeller as a force applied artificially into the computational domain. This in turn reduces the turn around time between design iterations, enabling airframes to be designed numerically rather than via experimental 'trial and error.'

This paper presents an implementation of an actuator disk and a blade element disk model in NUMECA FINE/Open using OpenLabs [2]. Three different loading profiles will be used and two different methods of calculating the lift and the drag for a blade element model are presented. The propellers have been implemented as a source term in the computational fluid dynamics code and the results obtained have been compared against those from XROTOR and from literature.

Matthew Anderson. The University of Sydney, Australia and Université Libre de Bruxelles, Belgium
KC Wong. The University of Sydney, Australia
Patrick Hendrick. Université Libre de Bruxelles, Belgium

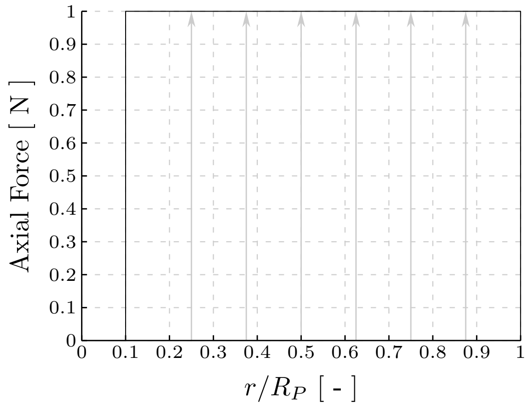

Fig. 1. Constant Axial Force

2. Propeller Models

2.1. Actuator Disks

Actuator disks are a simple way of modelling propellers that do not take into account the incoming flow, but instead add a constant force to the flow. This is implemented as a source in the OpenLabs code which is added to a prescribed set of cells. This is different to the pressure jump implementation where a pressure discontinuity is added between the cells to produce an infinitely thin disk.

Each for the three loading models considered here - constant, linear and NUMECA are described below. Figures 1, 2 and 3 show the loading models for half of the propeller using $R_p = 1$ [m], $R_h = 0.1R_p$, $T = 1$ [N] and $Q = 1$ [N.m].

Constant Loading The simplest method of approximating a propeller is as a constant thrust loading. This is modelled by

$$f_{ba} \quad = \quad T \tag{1}$$

where f_{ba} is the axial applied force, T is the desired thrust, R_p is the radius of the propeller and R_h is the hub radius. The radial and tangential forces, f_{br} and $f_{b\theta}$ respectively, of the propeller are both set to zero. The resulting thrust distribution is shown in Figure 1.

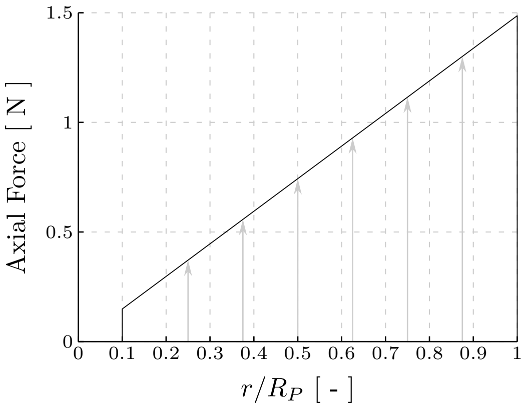

Fig. 2. Linear Axial Force

Linear Loading The linear loading model attempts to model the increasing thrust generated towards the tips of the propeller, however fails to model the loss of lift at the tips due to tip losses. The loading profile is modelled by

$$f_{ba} = \frac{3}{2}\,T\,\frac{R_p^2 - R_h^2}{R_p^3 - R_h^3} \times R_p \tag{2}$$

where f_{ba} is the axial applied force, T is the desired thrust, R_p is the radius of the propeller and R_h is the hub radius [3]. Both the radial and tangential forces, f_{br} and $f_{b\theta}$ respectively, of the propeller are set to zero. The resulting thrust distribution is shown in Figure 2.

NUMECA Loading The NUMECA loading case was sourced from the FINE/Marine Manual [4] and modified to reduce the required inputs to the thrust, torque and the disk thickness. This greatly simplifies the use of the model and removes the singularity present at zero velocity from the original implementation. This model provides a much more 'realistic' shape to the lift distribution compared to the previous two models, and is capable of providing an estimate of the applied tangential force.

The thrust and torque are calculated first by determining an auxiliary equation such that

$$aux = \frac{13.125}{\pi R_p^2}\frac{1}{\left(4+3\frac{R_h}{R_p}\right)(1.0 - R_h/R_p)}\frac{r/R_p - R_h/R_p}{1 - R_h/R_p} + \sqrt{1 - \frac{r/R_p - R_h/R_p}{1 - R_h/R_p}}\pi\left(R_p^2 - R_h^2\right) \tag{3}$$

where r is the current radial co-ordinate, R_p is the propeller radius and R_h is the hub radius. This auxiliary equation is then used to calculate the axial and tangential force such that

$$f_{ba} = T \times aux \tag{4}$$

$$f_{bt} = Q \times \frac{aux}{r} \tag{5}$$

where T is the thrust and Q is the torque to compute the applied forces. As before, the radial force fbr is assumed zero. Figure 3 shows the resulting force distribution.

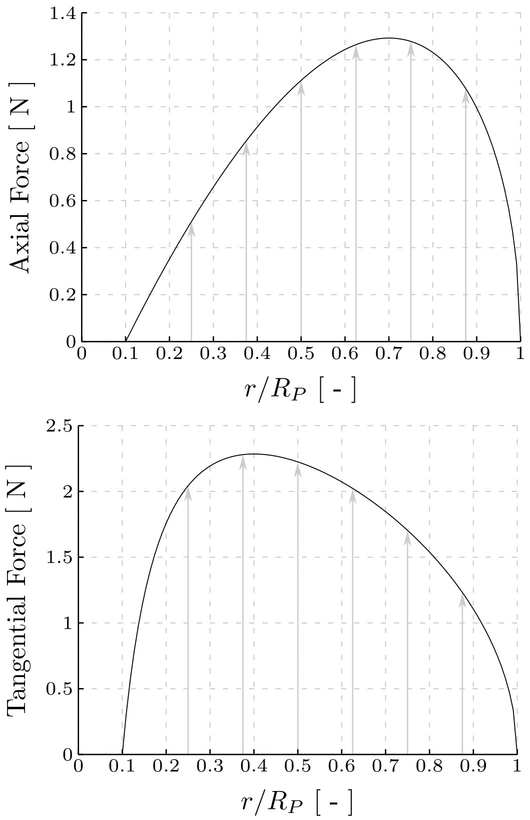

Fig. 3. NUMECA Axial Force and Tangentail Force Distribution

2.2. Blade Element Models

Blade element models (BEM) use the geometry of the propeller, the lift and drag characteristics of the aerofoil and information about the incoming flow to predict the performance of the propeller. Blade element models can provide a much better approximation to the flow than actuator disk models, however require a lot more information about the propeller and are more complex to implement.

In order to calculate the thrust and torque of a propeller in BEM, the lift and drag components of the aerofoil forces need to be rotated from the aerofoil axis into the propeller axis. This is done by

$$f_{ba} = 0.5\,\rho\,V_1^2\,c\,\left(R_p - R_h\right)\,\left(C_L cos\left(\phi\right) - C_D sin\left(\phi\right)\right) \times B \tag{6}$$

$$f_{bt} = 0.5\,\rho\,V_1^2\,c\,\left(R_p - R_h\right)\,\left(C_L sin\left(\phi\right) + C_D cos\left(\phi\right)\right) \times B \tag{7}$$

where ρ is the air density, $V1$ is the airspeed as seen by the propeller c is the propeller chord, C_L is the lift coefficient, C_D is the drag coefficient, ϕ is the blade pitch angle, and B is the number of blades (Figure 4) .

For a propeller with an axial direction of positive X, the quantities in Figure 4 are

$$V_0 = V_x \tag{8}$$

$$V_1 = \sqrt{V_0^2 + V_2^2} \tag{9}$$

$$V_2 = V_y \times \cos\left(\Phi\right) - V_z \times \sin\left(\Phi\right) + \Omega \times r \tag{10}$$

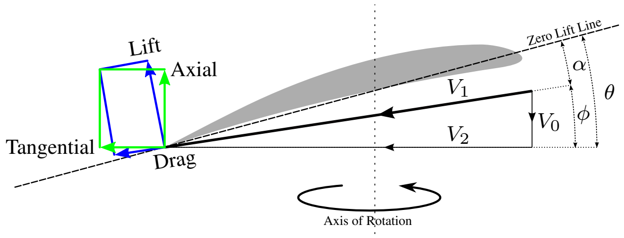

Fig. 4. Calculating the Local Angle of Attack for Each Blade Element.

where $V_{x/y/z}$ is the velocity component in X/Y/Z respectively, Ω is the propeller rotation speed in radians per second and Φ is the angle of the blade in the disk calculated via

$$\Phi = atan2\,(Y, Z) \tag{11}$$

The angle between the thrust and aerofoil lift directions can then be calculated via

$$\phi \quad = \quad atan2\,(V0, V2) \tag{12}$$

and the local angle of attack on the blade is then

$$\alpha \quad = \quad \theta - \phi \tag{13}$$

where θ is the blade pitch, or the angle between the disk and the zero lift line of the aerofoil. The $atan2()$ operator is used as it returns an angle between $-\pi$ and π, thus simplifying the lift and drag calculations when the angle of attack becomes negative.

2.2.1. *Lift and Drag Modelling*

Two different methods have been investigated for modelling the lift and the drag of the aerofoil. The first takes from O'Brien [3] which provides a linear approximation to the lift and a quadratic approximation to the drag. The second model from Traub [5] gives a better approximation post-stall. The results of each model are shown in Figure 5.

The first method uses

$$C_L \quad = \quad \begin{cases} C_{L_\alpha} \times (\alpha - \alpha_0), \\ \quad \hookrightarrow \alpha_{min} \leq \alpha \leq \alpha_{max} \\ C_{L_\alpha} \times (\alpha - \pi - \alpha_0), \\ \quad \hookrightarrow \alpha_{min} + \pi \leq \alpha \leq \pi \\ C_{L_\alpha} \times (\alpha + \pi - \alpha_0), \\ \quad \hookrightarrow -\pi \leq \alpha \leq -\pi + \alpha_{max} \\ C_{L_{\alpha_{stall}}} \times (\alpha - \alpha_{max}) + C_{L_{max}}, \\ \quad \hookrightarrow \alpha_{max} < \alpha < \pi - \alpha_{min} \\ C_{L_{\alpha_{stall}}} \times (\alpha - \alpha_{min}) + C_{L_{min}}, \\ \quad \hookrightarrow -\pi + \alpha_{max} < \alpha < \alpha_{min} \end{cases} \tag{14}$$

and

$$C_D \quad = \quad \begin{cases} \left(C_{D_0} + C_{D_1}\,\alpha + C_{D_2}\,\alpha^2\right), \\ \quad \hookrightarrow -\pi/2 \leq \alpha \leq \pi/2 \\ \left(C_{D_0} + C_{D_1}\,(\alpha - \pi) + C_{D_2}\,(\alpha - \pi)^2\right), \\ \quad \hookrightarrow \alpha > \pi/2 \\ \left(C_{D_0} + C_{D_1}\,(\alpha + \pi) + C_{D_2}\,(\alpha + \pi)^2\right), \\ \quad \hookrightarrow \alpha < -\pi/2 \end{cases} \tag{15}$$

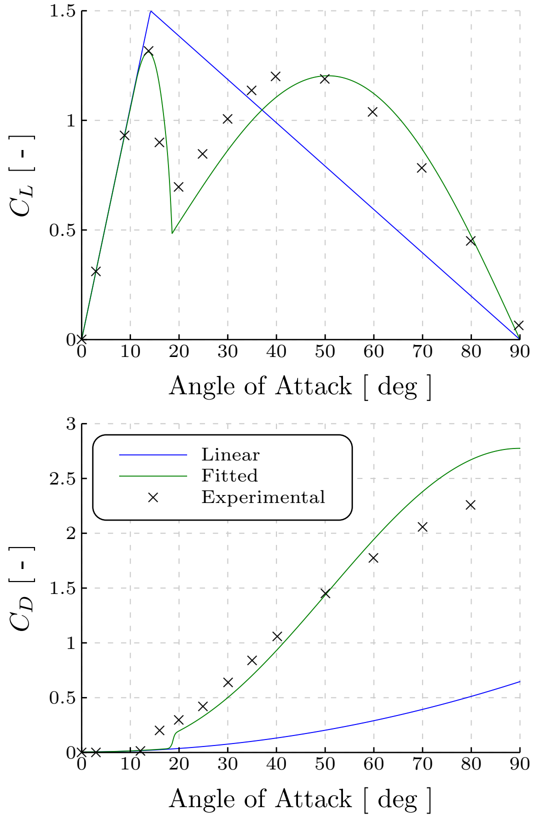

Fig. 5. Lift and Drag Modelling for the Blade Element Models using a NACA0012 Aerofoil. Data taken from [6].

The second method uses

$$\begin{aligned} C_L \quad &= \quad \frac{\gamma\pi}{2}\left(2 + \eta_1\left(\eta_2\sqrt{\beta} - 1\right)\right)^2 \sin(\alpha - \alpha_0)\,\eta_2 + \\ &\qquad (1 - \eta_2)\,B2\,\sin^a(\alpha)\,\cos(\alpha) \qquad (16) \\ C_D \quad &= \quad \left(C_{D_0} + \left(C_{L_\alpha}(\alpha - \alpha_0) - C_{L_{md}}\right)^2\,k_p\right)\,\eta_2 + \\ &\qquad (1 - \eta_2)\,(0.5\,C_{D_0}\,\cos(\alpha) + C_L\,\tan(\alpha)) \qquad (17) \end{aligned}$$

where $\gamma = C_{L\alpha}/(2\pi)$, β is a function that describes the coefficient of lift as a function of angle of attack between the loss of lift linearity (at angle α_i) and the post stall lift recovery (at angle α_v), α_0 is the angle of attack of zero lift, C_{Lmd} is the C_L at minimum drag and k_p is the induced drag constant (in the standard $C_D = C_{Do} + k_p\,C_L^2$ model). η_1 and η_2 are switching functions that are defined as

$$\eta_1 \quad = \quad (1 + \exp(-\omega \times (\alpha - \alpha_i)))^{-1} \tag{18}$$

$$\eta_2 \quad = \quad (1 + \exp(\quad \omega \times (\alpha - \alpha_v)))^{-1} \tag{19}$$

where ω is the severity of the switching function, α_i is the angle of attach at loss of lift linearity and α_v is the angle of attack at the post stall lift recovery.

2.3. Correction Factors

Tip Losses The lift losses due to the tip of the propellers are calculated using Prandtl's Tip Loss factor [7]. A 'correction' factor is calculated such that

$$f = \frac{2}{\pi} \cdot cos^{-1}\left(exp\left(\frac{-\left(B/2\right)\left(1-\frac{r}{R_P}\right)}{\left(\frac{r}{R_P}\right)\cdot sin\left(\phi\right)}\right)\right) \qquad (20)$$

where ϕ is the angle between the thrust and lift directions and B is the number of blades. f is then multiplied by the axial force fba in order to calculate the 'corrected' lift distribution (Figure 6).

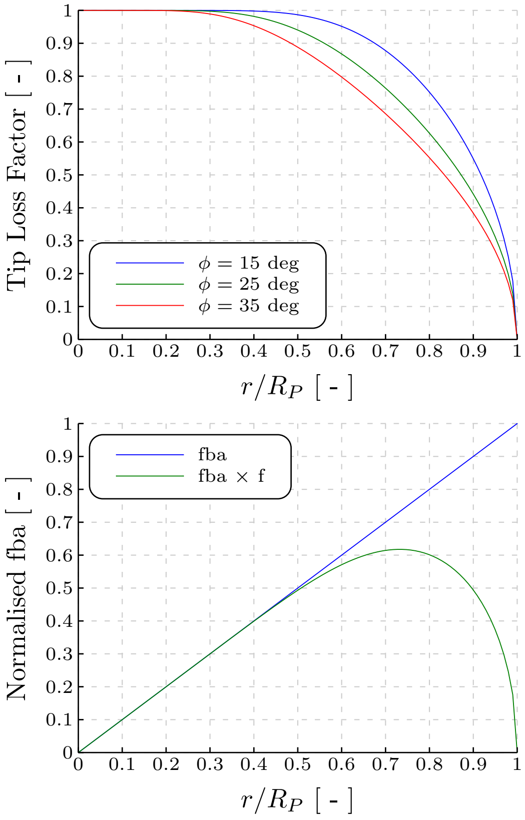

Fig. 6. Tip and Hub Loss Functions and Resulting Lift Distribution. B = 2, normalised fba calculated with $\phi = 15$ deg.

Mach Number The Mach number correction is calculated as

$$C_D = \frac{C_D}{1 - M^2} \qquad (21)$$

where M is the Mach number of the blade element and is applied to the calculated co-efficient of drag for each blade element. It is important to note that if incompressible flow is used for the simulation, the Mach correction term must be left out (or the Mach number artificially set to zero) otherwise it causes the flow solver to crash.

3. Implementation in OpenLabs

Propeller models are generally either implemented using a pressure jump discontinuity between cells or as a source term in the momentum equations for a set of cells [3]. In OpenLabs, the addition of a source term is utilised which adds the required force to the cells that are defined as being within the propeller. This is achieved simply by adding the force applied to the flow divided by the volume of the actuator disk. An example of adding axial force source term to the X momentum equation is shown below:

```
=>SOURCETERMS
# Axial Forces
@ SOURCE: sourceToMomX
->EXPRESSION: (fba) / (A*delta)
->AddToExistingPde: MomentumXEquation
```

where fba is the applied force, A is the area of the actuator disk, and δ is the thickness of the disk.

This introduces an issue as the propeller code assumes that the volume of the cells the force is applied to is equal to the volume of the disk prescribed. This is almost never the case as some cells will be partially inside the propeller disk, yet the whole volume contributes as FINE/Open uses cell-centred elements in its calculations. This results in a different amount of force actually being applied, and requires post-processing to calculate the exact volume of the propeller disk, which is then needed to be manually scaled and updated, and the calculation re-run.

4. Simulation Parameters

The mesh was made using NUMECA Hexpress v3.1-3, and consisted of 1 066 539 fully hexa-hedral cells.

The simulations were run in NUMECA FINE/Open v3.1-3 with OpenLabs. The boundary conditions were set as an inlet/outlet system with an incoming velocity prescribed as per the recommendations of O'Brien [3]. The external (a Riemann invariant far-field) boundary condition enforces a specified flow velocity at the boundary, causing an un-natural pressure rise at the boundary. The flow type was set to Euler as only the forces applied and the resulting flow field are of interest.

The simulations were conducted using a free stream velocity of 10 m/s using incompressible air as the fluid. An R_p of 0.1397 [m] (5.5 [in]) and a disk thickness of 0.02 [m] were used, and the propeller advance ratio was set at 0.6. The blade geometries were set using the data from Gray [8] using the 50 [deg] blade pitch propeller. Four multigrid levels and CPU booster were used to help accelerate convergence. CFView v9.0-3-5 was used for post-processing.

XROTOR [9] was used to compare the results from the blade element codes and calculate the resulting flow field from the propellers. The simulation set-up was verified against the results from Gray [8] and modified for the case presented here.

5. Results

The thrust is calculated by taking the scalar average of the fba force over the propeller disk. The torque is calculated in a similar manner however the scalar average is taken over $fbt \times r$, where r is the radial distance in the propeller disk.

5.1. Result Normalisation

All of the results have been normalised following [3] using

$$V_{normalised} = \frac{V_{axial} - V_{\infty}}{V_i} \tag{22}$$

where V_{axial} is the calculated axial speed of the flow and V_{∞} is the free stream velocity. V_i is the induced velocity and is calculated via

$$V_i = -\frac{V}{2} + \sqrt{\left(\frac{V}{2}\right)^2 + \frac{T}{2\rho A}} \tag{23}$$

where v_i is the induced velocity, V is the freestream velocity, T is the thrust produced, ρ is the density and A is the disk area.

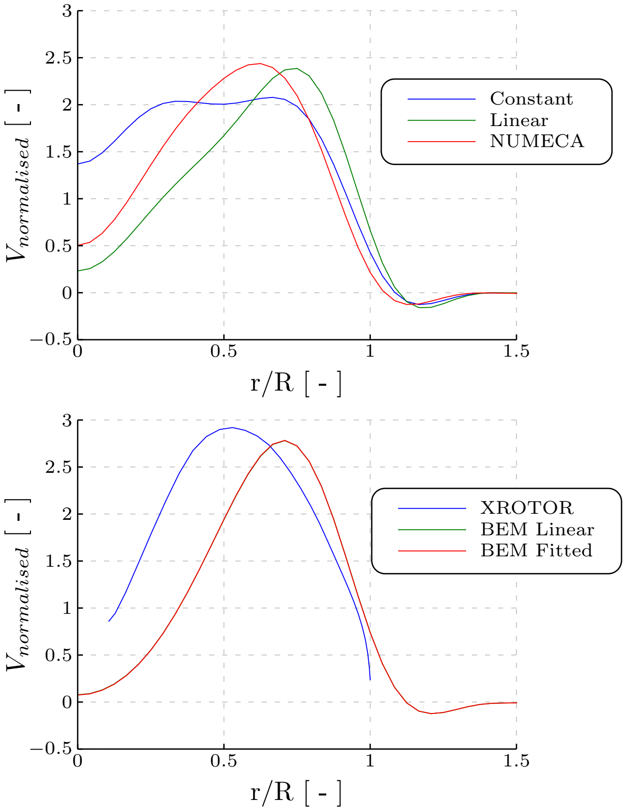

Fig. 7. A Comparison of the Axial Wake Velocities Far Downstream (Five Propeller Radii) of the Propeller Disk between XROTOR and FINE/Open. The actuator disk models are shown on top and the blade element models are shown on the bottom.

5.2. Flow Field

The resulting wake velocity profile is shown in Figures 7 and A-1 through B-5. Near the propeller disk, the flow is accelerated to the induced velocity and continues to accelerate to around twice the induced velocity far downstream as the propeller stream tube contracts. It was noted during the simulations that if the wake dissipated excessively, the cause was primarily an unrefined grid in the rotor wake.

The actuator disk models have very differing velocity profiles initially around the propeller, with each following more or less the shape of the loading profile. Further downstream, the linear and NUMECA loading cases start to become very similar with the highest speed of the flow being slightly more outboard in the linear case. The constant loading case starts very flat and dissipates somewhat by five radii downstream, however still has a very high induced velocity towards the centre of the disk compared to the other models.

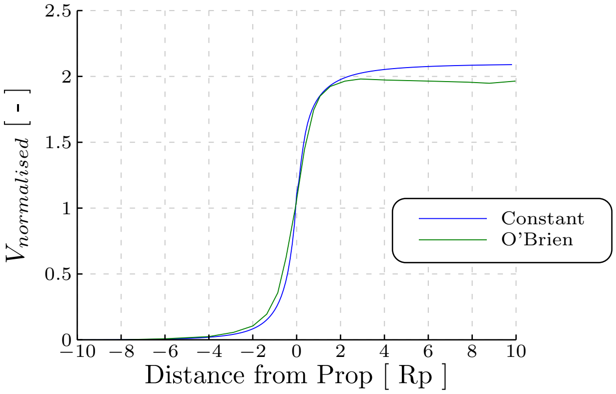

Fig. 8. The Normalised Induced Velocity Taken Through the Middle of the Thrust Loading.

The shape of the wake of the blade element models match well to the solution from XROTOR, as does the NUMECA loading case. The results of each of the OpenLabs blade element codes lie almost completely over each other as the propeller is operating in the linear lift region where both models overlap. A greater difference would be apparent after the propeller has stalled where the linear model will failed to model the lift and drag of the propeller as accurately as the fitted method.

5.3. Induced Velocity

O'Brien [3] measured the induced velocity of his wake simulations by taking a line through the centre of a constantly loaded actuator disk to verify the model was behaving correctly. This only works in limited cases as the disk must be constantly loaded and have no hub. A similar approach was taken here for the constant case , however the line was take through $r/Rp = 0.55$, half way between the hub and the tip of the rotor blade. The results are shown in Figure 8.

As can be seen, the results match each other well, verifying the implementation here. This also shows that implementing the propeller as a force source yields very similar results to the pressure implementation. Taking the normalised induced velocity through the centreline however doesn't work for the other loading cases and a different approach was taken.

Figure 9 presents the volume integral of the positive normalised induced velocity for each of the models presented. In front of the propeller, the induced velocity is much higher. This is because although the induced velocity through the centreline decreases further in front of the propeller, the stream tube affected by the propeller grows, meaning the integral reduces much more slowly down to zero at the boundary. The actuator models also don't reach the normalised induced velocity of two as before - this is because the formulation of Froude's theory neglects tip losses, which are present in the CFD solution. The BEM models don't suffer this problem because the

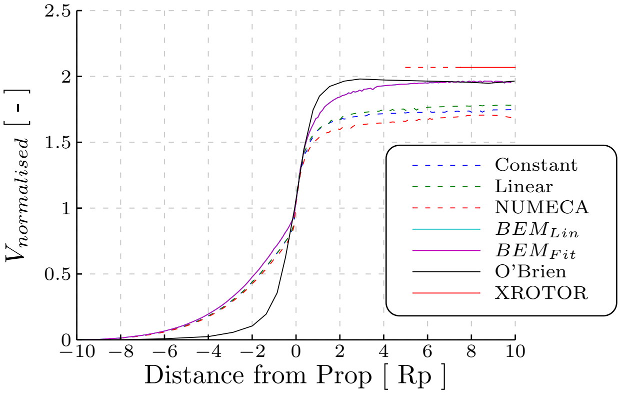

Fig. 9. The Volume Integral of the Normalised Induced Velocity Up and Downstream of the Propeller.

tip losses are included inherently in the BEM calculation and thus the thrust produced includes the loss factor. The XROTOR solution shown is valid far downstream and comes out just over two, a very reasonable result given the coarseness of the XROTOR grid.

5.4. Generated Thrust and Torque

A verification of the generated thrust and torque was run for the blade element codes using the data from Gray [8]. The 50 [deg] propeller was chosen and a sweep of the advance ratio from 0.5 to 2.5 was conducted. In this case, advance ratio is calculated via

$$J = \frac{V_\infty}{nD} \tag{24}$$

where J is the advance ratio, V_∞ is the free-stream velocity, n is the propeller rotation speed in revolutions per second and D is the propeller diameter. V_∞ was kept constant at 10 [m/s] and the propeller rotation speed was varied.

Figure 10 shows the results with the data from Gray [8] overlaid. The data matches more closely to the 45 [deg] data than the 50 [deg] data, the offset most likely due to the difference in definition and the aerofoil data used. Typically in practice, propeller pitch is measured from the pressure surface as opposed to the zero-lift line, resulting in a reference difference between the CFD solution (where the pitch is measured from the zero lift line) and the experimental data. This coupled with the unknown aerofoil data results in the offset of around 5 degrees between the predicted and actual performance pitches.

Taking this into account, the C_T in the linear region matches well with the experimental data for both the linear and fitted blade element models. Post blade stall, more of a difference is seen, with the fitted model dropping sharply and recovering somewhat, whereas the linear model only experiences a slight drop off in C_T. Gur and Rosen [10] noted that post blade stall modelling using blade element methods coupled with 2D aerofoil data greatly under-predicts the thrust produced as it fails to account for the Coriolis effect which delays boundary layer separation. Their method presented in [10] may help to improve the blade element results at stalled propeller conditions, though was not investigated for this work.

The C_Q data matches less well - the BEM solution under-predicts the torque of the propeller by around an order of magnitude. This is most likely due to poor modelling of the drag using 2D drag data to model the rotating, 3D propeller case. Nonetheless, the general shape of the curve for the fitted case resembles that of the experimental data and the magnitude of the drag force and easily be changed to match what is seen in practice.

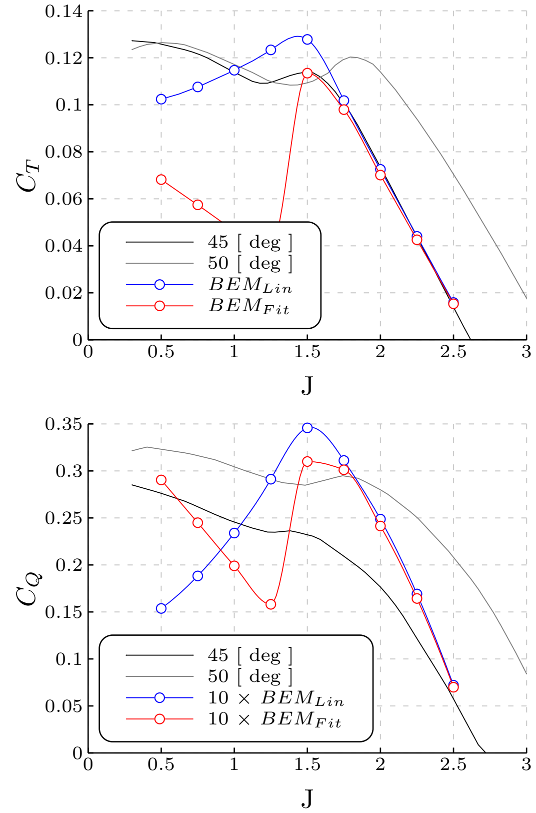

Fig. 10. A Comparison of the Predicted C_T and C_Q for the Blade Element Models versus the Data from Gray [8]. The blade element code has been run using the pitch profile for a propeller with 50 [deg] pitch.

5.5. Computation Time

A comparison of the computation time to run each model is presented in Figure 11. The computation time per iteration has been normalised against a case run without any propeller code implementation.

As to be expected, the simple propeller cases (constant and linear actuator disks) take less time to compute than the more complicated propeller models (such as the blade element models). Several optimisations were included in the more complicated codes such as using if statements to stop large calculations occurring outside the propeller disk. In a larger-scale simulation, partitioning the grid and using separate domains could further reduce the impact of the OpenLabs codes on the computation time.

The number of iterations to convergence has not been shown as it is very case/parameter specific. However, in these test cases, the actuator disk models converged in a similar number of iterations to each other, as did the blade element models, albeit in more iterations than the actuator disk models. The actuator disk models also proved to be more robust in other

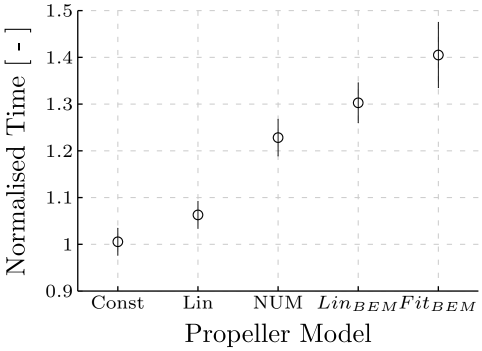

Fig. 11. Computation Time Comparison. The circles represent the average cycle computation time with the propeller model and the lines represent the standard deviation of the cycle times. The data has been normalised by the average cycle computation time without the propeller models activated.

testing when compared to the blade element models which often required the actuator disk models to provide the flow fields for the initial conditions.

5.6. Discussion

Each of the different models produces a different wake profile due to the way that the force is applied to the fluid volume (Figures A-1 through B-5). While the constant and linear loading profiles may not be the most realistic, they are easy to implement and use, have the least trouble in converging and for a quick idea of the flow are ideal. The NUMECA loading profile is almost equally as easy to implement, however unrealistic input values for the torque often cause convergence issues. This can easily be overcome by setting the torque to zero if the tangential component of the propeller wake is not of interest.

The blade element models offer the best idea of how the propeller will affect the flow as the force produced depends upon the incoming flow, however they are significantly more challenging to implement, take longer to converge and are more difficult to use than the actuator disk models. Also, as propellers are generally operated in un-stalled conditions, it is worth using only the linear model (unless stalled flow is expected) as generating the co-efficients for the fitted model equations can be difficult.

Overall, the best choice of propeller model to use rests with what the user is trying to analyse and which part of the flow is considered to be the most important. If the purpose of the model is simply to provide some propeller wash into the domain, then the actuator disk models are the preferred choice, with NUMECA model providing the closest force distribution to what one would expect. If the user is looking into how the propeller is affected by another body, the propeller forces are unknown or the flow is non-axial then the blade element models are the better choice as actuator disk models cannot model the effects of incoming flow.

6. Conclusion

This paper has presented the implementation for an actuator disk code with three differing loading profiles and a blade element code in NUMECA FINE/Open using OpenLabs. The resulting flow profiles were shown and compared to XROTOR and the computational cost of each of the models calculated.

The propeller models were shown to compare well against the results from literature and those from XROTOR. The best model to use is based on what the user wants to model - actuator disks are good for producing a propeller wake, however do not change their forces in different flow conditions making them unsuitable for modelling the effects of bodies on propellers.

7. Acknowledgements

A big thank you to Frank Buysschaert for the much needed support with fixing broken computers and FINE/Open (and code). Thank you also to Dries Verstraete for the expertise on propellers and a complied version of XROTOR for Windows and to NUMECA for their technical help in implementing these models using OpenLabs.

References

1. J. M. Moschetta. The Aerodynamics of Micro Air Vehicles: Technical Challenges and Scientific Issues. In 48^{th} *International Symposium of Applied Aerodynamics*, 2013.
2. NUMECA International. *User Manual: FINE/Open v3.1*, July 2014.
3. D. O'Brien. *Analysis of Computational Modeling Techniques for Complete Rotorcraft Configurations*. PhD thesis, Georgia Insititute of Technology, 2006.
4. NUMECA International. *User Manual: FINE/Marine v2*, March 2012.
5. L. W. Traub. High-Incidence Airfoil Lift and Drag Estimates. *Journal of Aircraft*, 49(1):311–314, 2012.
6. C. Critzos, H. Heyson, and R. Jr. Boswinkle. Aerodynamic Characteristics of NACA 0012 Airfoil Section at Angles of Attack from 0° to 180°. Technical Report 3361, NACA, January 1955.
7. E. Kulunk. *Aerodynamics of Wind Turbines, Fundamental and Advanced Topics in Wind Power*. InTech, 2011.
8. W. Gray. Wind-Tunnel Tests of Single- and Dual-Rotating Tractor Propellers at Low Blade Angles and of Two- and Three Blade Tractor Propellers at Blade Angles up to 65°. Technical report, Langley Memorial Aeronautical Laboratory, NACA, 1943.
9. M. Drela. XROTOR. `http://http://web.mit.edu/drela/Public/web/xrotor/`.
10. O. Gur and A. Rosen. Comparison between Blade-Element Models of Propellers. *The Aeronautical Journal*, 112:689–704, 2008.

Appendices

A. Induced Velocity Contours

The following figures show the wake profile for each of the propeller models normalised to the induced velocity.

$R_P = 0.1397$ [m], $R_h = 0.1 \times R_P$, $V = 10$ [m/s].

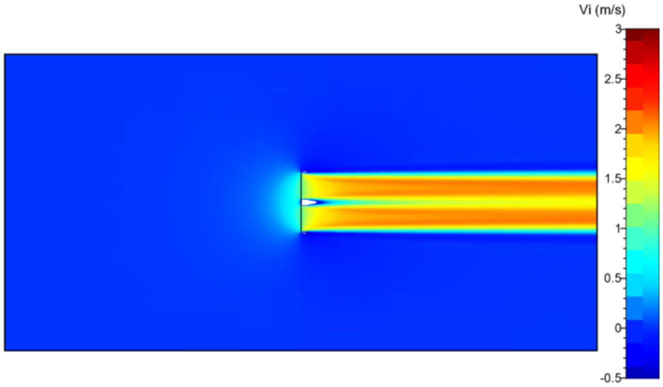

Fig. A-1. Normalised Induced Velocity for the Constant Actuator Disk Profile.

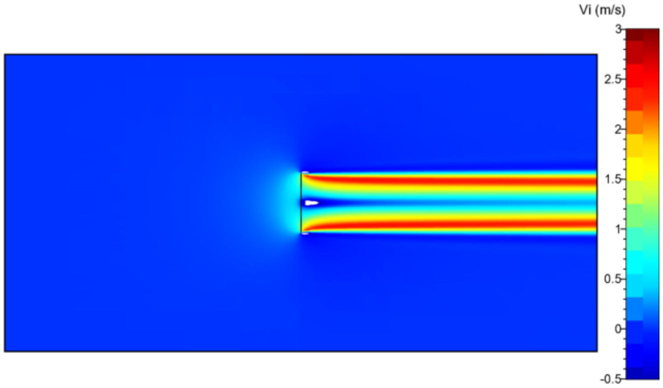

Fig. A-2. Normalised Induced Velocity for the Linear Actuator Disk Profile.

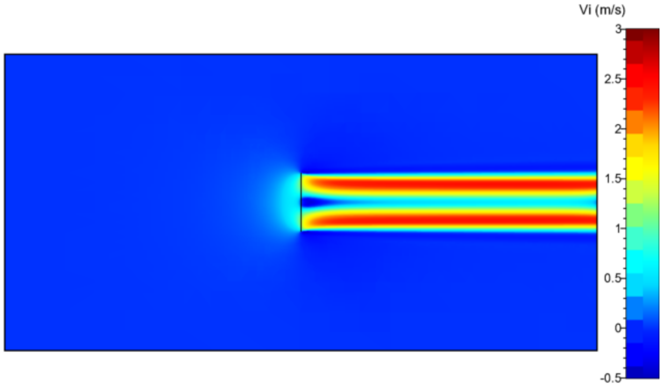

Fig. A-3. Normalised Induced Velocity for the NUMECA Actuator Disk Profile.

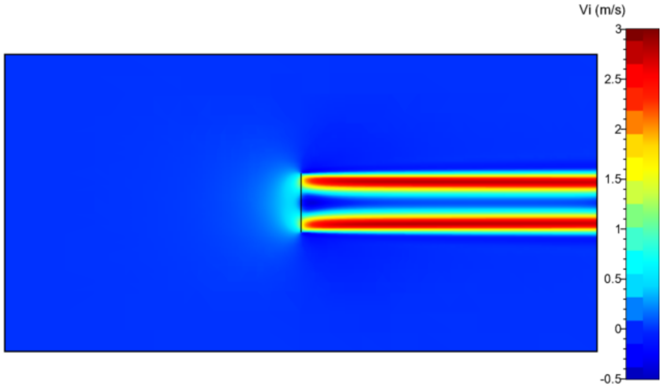

Fig. A-4. Normalised Induced Velocity for the Linear Blade Element Model.

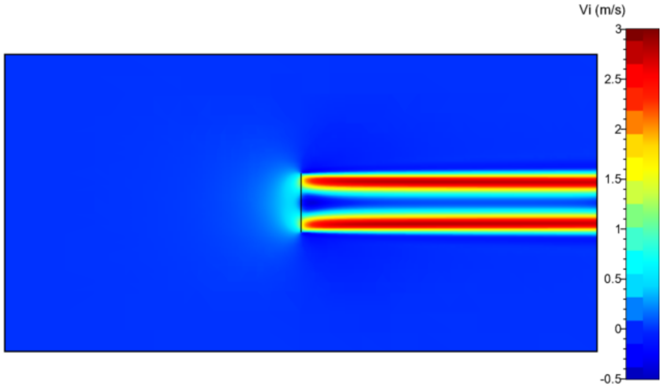

Fig. A-5. Normalised Induced Velocity for the Fitted Blade Element Model.

B. Wake Profiles in the Slipstream

The following figures show the wake profile for each of the models taken through the middle of the disk at various axial locations.

$R_P = 0.1397$ [m], $R_h = 0.1 \times R_P, V = 10$ [m/s].

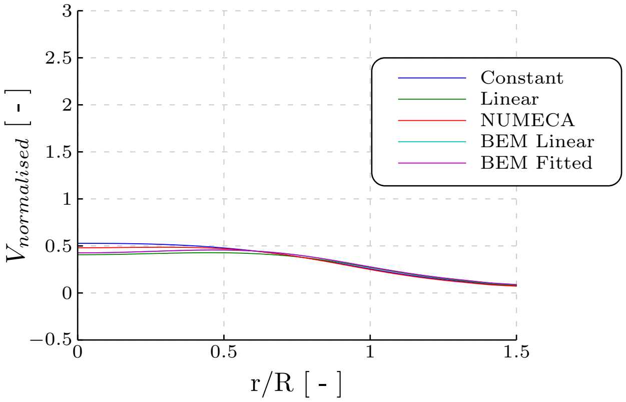

Fig. B-1. Normalised Velocities Half the Radii Upstream.

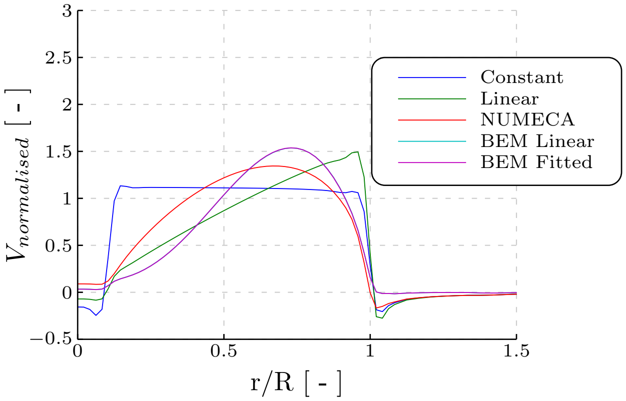

Fig. B-2. Normalised Velocities at the Centre of the Propeller Disk.

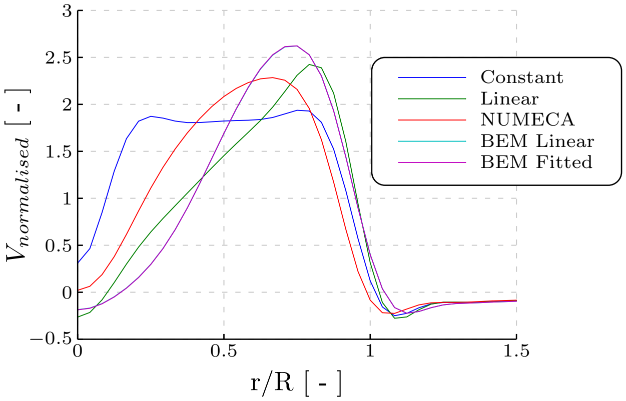

Fig. B-3. Normalised Velocities 1 Radius Downstream.

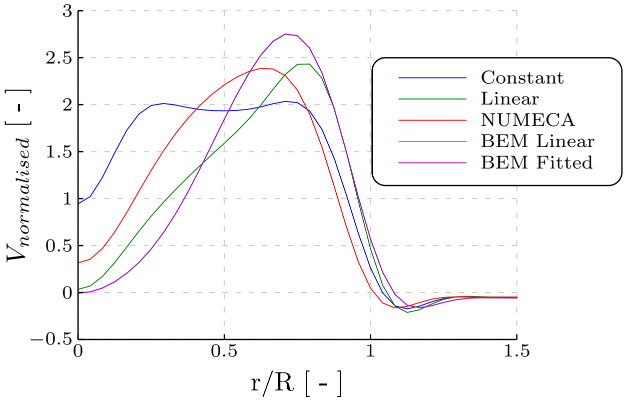

Fig. B-4. Normalised Velocities 2 Radii Downstream.

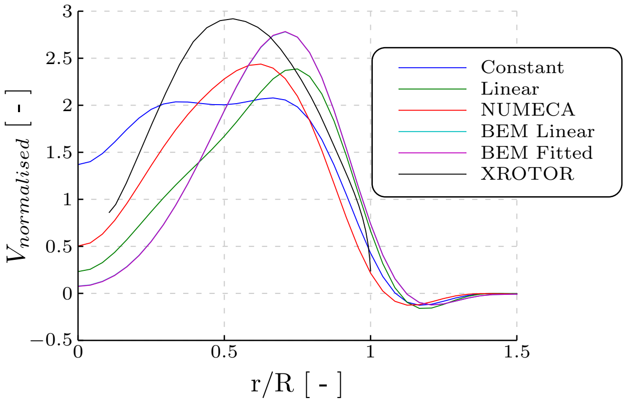

Fig. B-5. Normalised Velocities 5 Radii Downstream.