DCFoil Developer and User Documentation

Galen W. Ng

April 10, 2024

Summary

DCFoil is a program for the dynamic analysis and design optimization of composite hydrofoils.

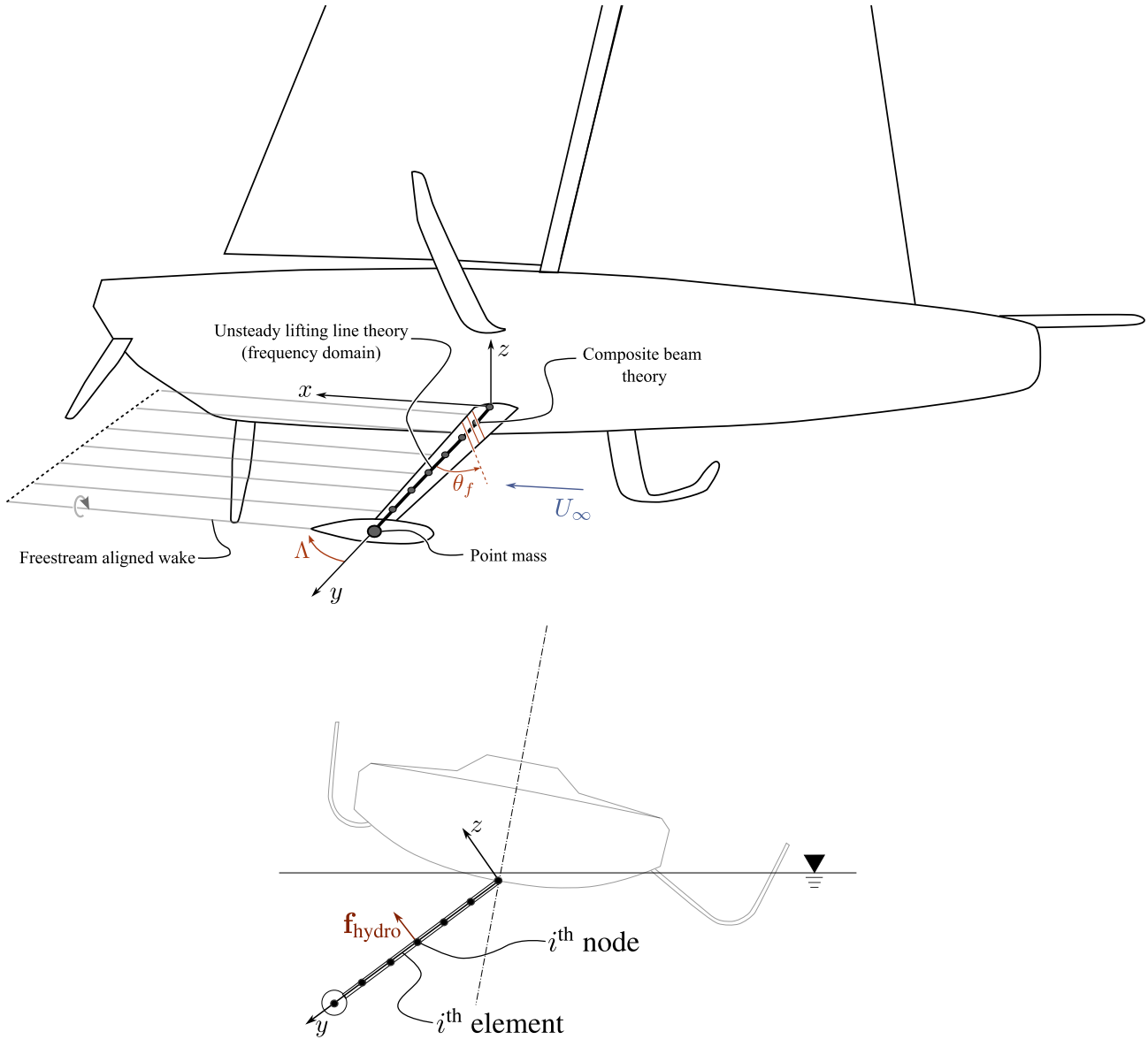

Figure 1: DCFoil modeling approach for an appendage

Contents

1 Coordinate system

The appendage coordinate system is flow is in the x-direction, span is in the y-direction, and the vertical direction is z.

2 Discretization

2.1 Structural model

The local beam model uses the spanwise direction as x (subscript 1), the chordwise direction as y (subscript 2), and the vertical direction as z (subscript 3). It is transformed to the global coordinate system by the rotation matrices. The rotation matrices are

$$\mathbf{T} = \tag{1}$$

2.1.1 Beam finite element

The composite beam model uses the well-known slender beam parameters EI_s, GJ_s, EA_s, and additionally S_s and K_s to account for structural warping of non-circular cross sections and material bend-twist coupling, respectively. No axial coupling to other degrees of freedom (DOF) is currently considered. The beam is discretized into 2-noded elements with 9 displacement DOFs denoted u, v, w, ϕ, θ, ψ, ϕ', θ', ψ'.

2.1.2 Beam parameters for composite materials via CLT

The beam parameters are computed from classical lamination theory (CLT) for composite plates using the high aspect ratio plate model from Weisshaar and Foist [1].

$$EI = c\left(D_{11} - \frac{D_{12}^2}{D_{22}}\right), \quad GJ = 4c\left(D_{66} - \frac{D_{26}^2}{D_{22}}\right), \quad K = 2c\left(D_{16} - \frac{D_{26}D_{12}}{D_{22}}\right) \tag{2}$$

These relations do not restrict chordwise rigidity (camber), but they do assume zero chordwise moment. Lottati [2] used the chordwise rigid relations, which simplifies the algebra.

The flexural stiffnesses D_{ij} (6x6 matrix) come via CLT

$$D_{11} = Q_{11}m^4 + 2\left(Q_{12} + 2Q_{66}\right)n^2m^2 + Q_{22}n^4 \tag{3}$$

$$D_{22} = Q_{11}n^4 + 2\left(Q_{12} + 2Q_{66}\right)n^2m^2 + Q_{22}m^4 \tag{4}$$

$$D_{66} = \left(Q_{11} + Q_{22} - 2Q_{12} - 2Q_{66}\right)n^2m^2 + Q_{66}\left(n^4 + m^4\right) \tag{5}$$

$$D_{12} = \left(Q_{11} + Q_{22} - 4Q_{66}\right)n^2m^2 + Q_{12}\left(n^4 + m^4\right) \tag{6}$$

$$D_{16} = \left(Q_{11} + Q_{22} - 2Q_{66}\right)nm^3 + \left(Q_{12} - Q_{22} + 2Q_{66}\right)n^3m \tag{7}$$

$$D_{26} = \left(Q_{11} + Q_{22} - 2Q_{66}\right)n^3m + \left(Q_{12} - Q_{22} + 2Q_{66}\right)nm^3 \tag{8}$$

where

$$m = \cos\theta_f, \quad n = \sin\theta_f. \tag{9}$$

The reduced in-plane stiffness coefficients Q_{ij} for the individual plies are

$$Q_{11} = \frac{E_1}{1 - \nu_{12}\nu_{21}}, \quad Q_{12} = \frac{\nu_{12}E_2}{1 - \nu_{12}\nu_{21}}, \quad Q_{22} = \frac{E_2}{1 - \nu_{12}\nu_{21}}, \quad Q_{66} = G_{12} \tag{10}$$

The following are numerical methods for computing structural properties of airfoils. The sectional mass properties are

$$m_s = \int_0^c \rho\left(Z_u - Z_\ell\right)dx \doteq \text{kg-m}^{-1} \tag{11}$$

$$I_s = \int_0^c \int_0^{t_{\max}} r^2 dm \tag{12}$$

where we evaluate the integrals numerically.

The sectional geometric properties are[1]

$$\bar{z} = \frac{1}{A}\int_0^c \frac{1}{2}\left[Z_u^2 - Z_\ell^2\right] dx \tag{13}$$

$$I = \int_0^c \frac{1}{3}\left[(Z_u - \bar{z})^3 - (Z_\ell - \bar{z})^3\right] dx \tag{14}$$

$$J = \int \tag{15}$$

$$\tag{16}$$

and torsional warping resistance due to tension from Lottati [2] is

$$S = \int_{-b}^{b} D_{22} x_a^2 dx, \quad \text{where } x_a = x - ba \tag{17}$$

2.1.3 Structural damping

We use a proportional (Rayleigh) damping model, which is commonly used for multi-DOF structures because of mathematical conveniences

$$\mathbf{C_s} = \alpha \mathbf{M_s} + \beta \mathbf{K_s}.$$

The mass proportional damping decreases with increasing response frequency whereas the stiffness proportional damping increases. We specify $\zeta = 2\%$ damping ratio at an undamped, in-vacuum, natural frequency giving $\beta = 2\zeta/\omega_{\max}$ where $\omega_{\max}$ is the highest undamped, in-vacuum, natural frequency in radians per second determined from modal analysis–in this paper, the fourth natural mode.

A few options are available but we typically use the stiffness proportional method ($\alpha = 0$) because response frequencies in water tend to be on the lower side; we want to avoid artificial overdamping of critical hydroelastic modes with $\omega < \omega_{\max}$. Structural damping is typically small compared to hydrodynamic damping so its main role is as a margin of stability if fluid damping is close to zero, thus accuracy of the structural damping model is not critical.

2.2 Hydrodynamic loads

2.2.1 Steady lifting line

The lifting line model derives from Glauert [3, Ch. XI] and works for arbitrary chord. Specifically, we are after sectional lift slopes ($c_{\ell_\alpha} = \mathrm{d}c_\ell/\mathrm{d}\alpha$). We assume

- the chord is small compared to the span,
- the wing is symmetric about the centerline,
- span is straight and orthogonal to the freestream
- trailing vortices are shed from the trailing edge and align with the freestream (no sweep or dihedral)

The wing is represented by superimposing "horseshoe" systems of vortex lines (analagous to a wire with electrical current). This is because the circulation across a wing is not constant. The free vortex system is a sheet of trailing vortices springing from the trailing edge. The induced velocity of an element of the line (ds) at point P from one vortex line of constant strength Γ is

$$dq = \frac{\Gamma}{4\pi r^2}\sin(\theta) ds \tag{18}$$

but in practice, one would solve this as an integral over the entire vortex line, so we will build up to the full wing.

[1]The torsional constant is NOT the same as the polar moment

To begin solution, we first assume the circulation is the Fourier series[2]

$$\Gamma(y) = 2U_\infty s \sum_{n=1}^{\infty} a_n \sin(n\theta) \quad \text{where } y = -\frac{s}{2}\cos(\theta) \quad \text{and } \theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]. \tag{19}$$

The difficulty is now determining the Fourier coefficients a_n so we need some relations for $\Gamma(y)$ to solve it.

One relation is the equation for the normal induced velocity (downwash velocity) at a point along the span

$$w(y) = \frac{1}{4\pi}\int_{-s/2}^{s/2} \frac{\frac{\mathrm{d}\Gamma}{\mathrm{d}\eta}}{y-\eta} d\eta = \boxed{-U_\infty \sum_{n=1}^{\infty} \frac{n a_n \sin(n\theta)}{\sin(\theta)}} \tag{20}$$

where η is the spanwise coordinate and s is total span. We skipped a few steps in the derivation [4, Sec. 3.7].

The second relation is from sectional lift as a function of circulation. Recall that the circulation at a section (derived from Kutta-Joukowski lift theorem) is

$$\Gamma(y) = \frac{1}{2} c_\ell c U_\infty = \frac{1}{2} a_0 c \left(U_\infty \alpha - w(y)\right) \tag{21}$$

where we made use of $c_\ell = a_0 \alpha_{\text{eff}} = a_0 (\alpha - w/U_\infty)$. After substitution of the Fourier series form and combining Equations (20) and (21), we end up with

$$\sum_{n=1}^{\infty} a_n \sin(n\theta)\left(n\mu + \sin(\theta)\right) = \mu\alpha \sin(\theta) \quad \text{where } \mu(\theta) = \frac{a_0 c(\theta)}{4s} \tag{22}$$

where the impact of geometric twist can be captured via $\alpha = \alpha_0 + f(\theta)$ where $f(\theta)$ is the twist as a function of the spatial variable.

Here's the digestion of the Julia code which does the numerical solution of Equation (22) but symmetrically about the centerline.

$\tilde{y} = \left[0, \frac{\pi}{2}\right]$ of size nNodes

$\mathbf{n} = [1 : 2 : 2 \times \text{nNodes}]$

$\mathbf{c} = c \sin(\tilde{y})$ (parametrized vector leading to elliptical planform)

$\mathbf{b} = \frac{\pi}{4}\frac{\mathbf{c}}{s/2}\alpha \sin(\tilde{y})$ (RHS of Equation (22) in vector form)

$\tilde{y}n = \tilde{y} \otimes \mathbf{n}$ (outer product)

$\mathbf{A_0} = \begin{bmatrix} | & | & \\ \sin(\tilde{y}) & \sin(\tilde{y}) & \cdots \\ | & | & \end{bmatrix}$ (square matrix of $\sin(\tilde{y})$)

$\mathbf{A_1} = \frac{\pi}{4}\frac{\mathbf{c}}{s/2} \otimes \mathbf{n}$ (outer product representing $n\mu$ on LHS)

$\mathbf{A} = \sin(\tilde{y}n) \odot (\mathbf{A_0} + \mathbf{A_1})$

$\mathbf{Ax} = \mathbf{b}$ (solve linear system for $\mathbf{x} = a_n$)

$$\Gamma(y) = 4U_\infty s/2 \left(\underbrace{\sin(\tilde{y}n)\mathbf{x}}_{\text{mat-vec product}} \right)$$

$$c_\ell = \frac{2\Gamma(y)}{U_\infty c}$$

$$c_{\ell_\alpha} = \frac{c_\ell}{\alpha}$$

[2] Kerwin and Hadler [4] use $\tilde{y}$ as θ

2.2.2 Flap effects

The effect of a flap comes from thin airfoil theory and is outlined in Glauert [5, 6]. A parameter $\phi = \cos^{-1}\left(\frac{c_f}{2} - 1\right)$ is a parametric variable for flap angle accounting for the length of flap chord c_f The lift slope ratio is

$$\frac{c_{\ell_\delta}}{c_{\ell_\alpha}} = 1 - \frac{\phi - \sin(\varphi)}{\pi} \tag{23}$$

and the change in moment (positive nose up) is

$$c_{mac_\delta} = -\frac{1}{4}\sin(\phi)\,(1 - \cos(\phi)) \tag{24}$$

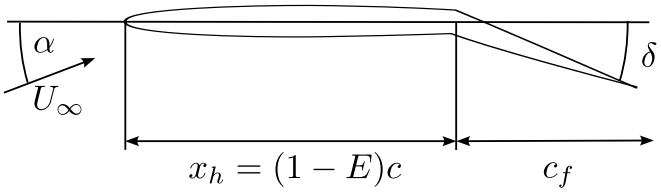

Figure 2: Flap diagram

2.2.3 Extension to unsteady frequency domain

We are interested in the sectional lift and moments for a harmonically oscillating body. Theodorsen [7] came up with the Theodorsen function $C(k)$ to account for the lag and deficit in forces with a farfield boundary condition. It is applied as a transfer function to the static hydrodynamics.

$$\begin{Bmatrix} F_z \\ M_y \end{Bmatrix}_i = -\left(\left[\mathbf{m_f}\begin{Bmatrix} \ddot{w} \\ \ddot{\psi} \end{Bmatrix}\right]_i + \left[\mathbf{c_f}\begin{Bmatrix} \dot{w} \\ \dot{\psi} \end{Bmatrix}\right]_i + \left[\mathbf{k_f}\begin{Bmatrix} w \\ \psi + \alpha_0 \end{Bmatrix}\right]_i + \left[\hat{\mathbf{c}}_\mathbf{f}\begin{Bmatrix} \dot{w}' \\ \dot{\psi}' \end{Bmatrix}\right]_i + \left[\hat{\mathbf{k}}_\mathbf{f}\begin{Bmatrix} w' \\ \psi' \end{Bmatrix}\right]_i\right)\Delta y_i \tag{25}$$

where Δy_i is the strip width at node i, which we assume to be equal to element length.

$$\mathbf{m_f} = \pi\rho_f b^2 \begin{bmatrix} 1 & ab \\ ab & b^2\left(\frac{1}{8} + a^2\right) \end{bmatrix} \tag{26}$$

$$\mathbf{c_f}(k) = \frac{1}{2}\rho_f b U_0 \left(\cos(\Lambda)\begin{bmatrix} c_{\ell_\alpha} 2C(k) & -b\left[2\pi + c_{\ell_\alpha}(1-2a)C(k)\right] \\ c_{\ell_\alpha} eb2C(k) & \frac{b}{2}(1-2a)(2\pi b - c_{\ell_\alpha} 2ebC(k)) \end{bmatrix}\right) \tag{27}$$

$$\mathbf{k_f}(k) = \frac{1}{2}\rho_f U_0^2 \cos(\Lambda)\left(\cos(\Lambda)\begin{bmatrix} 0 & -C(k)2bc_{\ell_\alpha} \\ 0 & -2eb^2 c_{\ell_\alpha} C(k) \end{bmatrix}\right) \tag{28}$$

$$\hat{\mathbf{c}}_\mathbf{f}(k) = \frac{1}{2}\rho_f b U_0 \sin(\Lambda)\left(\begin{bmatrix} 2\pi b & 2\pi ab^2 \\ 2\pi ab^2 & 2\pi b^3\left(\frac{1}{8} + a^2\right) \end{bmatrix}\right) \tag{29}$$

$$\hat{\mathbf{k}}_\mathbf{f}(k) = \frac{1}{2}\rho_f b U_0 \sin(\Lambda)\left(U_0\cos(\Lambda)\begin{bmatrix} c_{\ell_\alpha} 2C(k) & -c_{\ell_\alpha} b(1-2a)C(k) \\ 2ebc_{\ell_\alpha} C(k) & \pi b^2 - c_{\ell_\alpha} eb^2(1-2a)C(k) \end{bmatrix}\right) \tag{30}$$

The extra $\hat{\square}$ matrices account for sweep effects on the quasi-steady (damping and stiffness) aerodynamics and are lumped into their respective global matrices if they are in phase with velocity or displacements.

2.2.4 Free surface effect via desingularized potentials

The *indirect method* solves for the strengths of the singularities σ. The perturbation potential is φ. The total velocity potential with steady forward speed ($\Phi = U_\infty x + \varphi$) is some integral of a source distribution $\sigma(\mathbf{x}_s)$ over the surface Ω outside the problem domain

$$\boxed{\Phi(\mathbf{x}) = \iint_\Omega \sigma(\mathbf{x}_s) G(\mathbf{x}) d\Omega} \tag{31}$$

$$\Phi(\mathbf{x}) = \iint_\Omega \sigma(\mathbf{x}_s) \frac{1}{\mathbf{x} - \mathbf{x}_s} d\Omega. \tag{32}$$

We compute the free surface effect using the method of de-singularization first applied in Cao et al. [8]. The boundary conditions give rise to the following boundary integral equations (these are also Fredholm integrals of the first kind)

$$\iint_\Omega \sigma(\mathbf{x}_s) \tfrac{1}{|\mathbf{x}_c - \mathbf{x}_s|} \mathrm{d}\Omega \qquad = \phi_0(\mathbf{x}_c) \quad \text{(Dirichlet) } \Gamma_d \tag{33}$$

$$\iint_\Omega \sigma(\mathbf{x}_s) \tfrac{\partial}{\partial n} \left(\tfrac{1}{|\mathbf{x}_c - \mathbf{x}_s|} \right) \mathrm{d}\Omega \quad = \chi(\mathbf{x}_c) \quad \text{(Neumann) } \Gamma_n \tag{34}$$

The fully nonlinear dynamic and kinematic boundary conditions for the free surface (derived from the unsteady Bernoulli equation) are

$$\underbrace{\tfrac{\partial \phi}{\partial t} + \frac{1}{2} |\nabla \Phi|^2}_{\approx U \frac{\partial \phi}{\partial x}} + g z_f \quad = -\frac{1}{\rho} (p - p_{\text{atm}}) \quad \text{(dynamic)} \tag{35}$$

$$\tfrac{\mathrm{D}\mathbf{x}_f}{\mathrm{D}t} \qquad = \nabla \Phi \quad \text{(kinematic)} \tag{36}$$

Fully nonlinear free-surface formulations have problems with breaking waves. As such, linearizing and using a body-exact method where the body boundary condition uses the updated wave elevation is better. The linearized dynamic boundary condition is

$$\frac{\partial \phi}{\partial t} + U_\infty \frac{\partial \phi}{\partial x} + g z = \underbrace{-\frac{1}{\rho} (p - p_{\text{atm}})}_{=0 \text{ for } z=\eta} \tag{37}$$

$$\eta = -\frac{1}{g} \left(\frac{\partial \phi}{\partial t} + U_\infty \frac{\partial \phi}{\partial x} \right) \tag{38}$$

which you can use after you solve for ϕ to determine wave elevation [9]. The linearized kinematic boundary condition is

$$\frac{\partial \phi}{\partial z} - \frac{\partial \eta}{\partial t} - U \frac{\partial \eta}{\partial x} = 0 \tag{39}$$

$$\frac{\partial \eta}{\partial t} = \frac{\partial \phi}{\partial z} - U \frac{\partial \eta}{\partial x} \tag{40}$$

Combining them gives the unsteady linearized free surface boundary condition

$$\boxed{U_\infty^2 \frac{\partial^2 \phi}{\partial x^2} + g \frac{\partial \phi}{\partial z}} = -\left(\frac{\partial^2 \phi}{\partial t^2} + 2 U_\infty \frac{\partial^2 \phi}{\partial x \partial t} \right) \tag{41}$$

which we then need to use to set up the LHS and RHS on Γ_d.

The linear system to solve comes from the mixed boundary conditions originally in Equation (33). The result from this are the unknown source strengths σ_i. For the linearized free surface boundary condition, we plug Equation (33) in to get

$$U_\infty^2 \iint_\Omega \frac{\partial^2}{\partial x^2} \left(\sigma(\mathbf{x}_s) \frac{1}{|\mathbf{x}_c - \mathbf{x}_s|} \mathrm{d}\Omega \right) + g \iint_\Omega \frac{\partial}{\partial z} \left(\sigma(\mathbf{x}_s) \frac{1}{|\mathbf{x}_c - \mathbf{x}_s| \, \mathrm{d}\Omega} \right) = 0 \quad \text{(Free surface Dirichlet BC)} \tag{42}$$

Looks pretty nasty, right? We need to evaluate the above equation in each free surface panel. But you will also have one more equation for the Neumann BC which will represent the body in the flow, that way it's not just solving a null space problem.

On a quad panel, the idea is to use shape functions $N_i(\xi, \eta)$ in a reference element at the four nodes

$$\mathbf{x} \approx \sum_{i=1}^{4} N_i(\xi, \eta)\mathbf{x}_i \qquad \frac{\partial \Phi}{\partial n} \approx \sum_{i=1}^{4} N_i(\xi, \eta)\frac{\partial \Phi}{\partial n}_i$$

so we can actually evaluate stuff like Equation (33). TODO: I think

We can write the above in matrix form as

$$\begin{bmatrix} A_{11} \cdots A_{1N} \\ \vdots \ddots \vdots \\ A_{N1} \cdots A_{NN} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \vdots \\ \sigma_N \end{Bmatrix} = \begin{Bmatrix} b_1 \\ \vdots \\ b_N \end{Bmatrix} \tag{43}$$

where $N = N_F + N_B$. From this, we are saying the Dirichlet and Neumann BCs are stacked vertically on the RHS.

Algorithm 1 is the overall procedure for computing the influence coefficients for a steady-state free surface.

Algorithm 1: Computation of hydrodynamic loads with a steady free surface boundary condition and constant forward speed

Data: Grid points (paneling of geometry) and flow speed
Result: Source strengths σ_i

1. Pass in (or generate) mesh and panel with node connectivity of control points $\mathbf{x}_c$ and source points $\mathbf{x}_s$;
2. Form $\mathbf{A}$ matrix from Equation (43);
3. Form $\mathbf{b}$ of the matrix equation (43) (i.e., Dirichlet on free-surface Equation (41) and Neumann on body);
4. Solve $\boldsymbol{\sigma} = \mathbf{A}^{-1}\mathbf{b}$;
5. Solve for the velocity potential Φ in each panel using numerical integration on Equation (31);
6. Solve for wave elevations from the dynamic condition using $\Phi(\mathbf{x})$ in Equation (39);
7. Solve for hydrodynamic loads on the body using Bernoulli's equation and integration of pressures;

2.2.5 Test problems

The first test problem is a dipole beneath a flat plane with $\varphi = 0$ and strength of $\mu = 1$. This corresponds to infinite Froude number (impulsive start) for the dipole. The dipole is at $\mathbf{x}_0 = [0, 0, -1]$. We find $\frac{\partial \varphi}{\partial n}$ on $z = 0$.

The first step is to convert the potential Equation (31) to a summation with one dipole term

$$\varphi(\mathbf{x}_f) = \underbrace{\frac{-\mu}{4\pi} \frac{x}{\left|\mathbf{x}_f - \mathbf{x}_0\right|^{3/2}}}_{\text{dipole}} + \sum_{j=1}^{N_F} \frac{\sigma_j}{\left|\mathbf{x}_f - \mathbf{x}_{sj}\right|} \tag{44}$$

where $\mathbf{x}_{sj}$ are the source points outside the domain.

If one plugs in the boundary condition $\varphi = 0$ on $z = 0$, we get the exact solution

$$\varphi(x, y, z) = \frac{-\mu}{4\pi} \frac{x}{\left[x^2 + y^2 + (z+1)^2\right]^{3/2}} + \frac{\mu}{4\pi} \frac{x}{\left[x^2 + y^2 + (z-1)^2\right]^{3/2}} \tag{45}$$

which gives the exact normal velocity on the plane $z = 0$ as

$$\left.\frac{\partial \varphi}{\partial n}\right|_{\text{exact}} = \frac{3}{2\pi} \frac{x}{\left(x^2 + y^2 + 1\right)^{5/2}} \tag{46}$$

The wave elevation is zero (math checks out).

Now, in the *numerical solution*, we solve a linear system for the unknown σ_j, and set up integrals to numerically evaluate φ and derived quantities from φ like $\frac{\partial \varphi}{\partial n}$. First, the linear system is

$$\sum_{j=1}^{N_F} \frac{\sigma_j}{\left|\mathbf{x}_i - \mathbf{x}_{sj}\right|} = \frac{x}{4\pi \left|\mathbf{x} - \mathbf{x}_0\right|^{3/2}} \tag{47}$$

The normal velocity is

$$\left.\frac{\partial \varphi}{\partial n}\right|_{\text{numerical}} = \iint_\Omega \frac{\partial}{\partial n} \left(\frac{\sigma(\mathbf{x}_s)}{|\mathbf{x}_c - \mathbf{x}_s|} + \frac{x}{4\pi \left|\mathbf{x} - \mathbf{x}_0\right|^{3/2}} \right) \mathrm{d}\Omega \tag{48}$$

determined from numerical integration. TODO: GGGGGGGGGGGGGGGGGGGGGGGG code this up Check **?**] thesis Appendix B for how to do normal velocity...

Now if we do a dipole moving in steady forward speed, the linear system is

$$U_\infty^2 \iint_\Omega \frac{\partial^2}{\partial x^2} \left(\sigma(\mathbf{x}_s) \frac{1}{|\mathbf{x}_c - \mathbf{x}_s|} \mathrm{d}\Omega + \frac{x}{4\pi \left|\mathbf{x} - \mathbf{x}_0\right|^{3/2}} \right) + g \iint_\Omega \frac{\partial}{\partial z} \left(\sigma(\mathbf{x}_s) \frac{1}{|\mathbf{x}_c - \mathbf{x}_s|\, \mathrm{d}\Omega} + \frac{x}{4\pi \left|\mathbf{x} - \mathbf{x}_0\right|^{3/2}} \right) = \mathbf{0} \tag{49}$$

3 Static solution

The static hydrostructural equation is

$$\mathbf{K}_{ss}\mathbf{u}_s = \mathbf{f}_{\text{hydro}} = \mathbf{K}_f\mathbf{u}_s \tag{50}$$

$$\mathbf{r}(\mathbf{u}_s) = \left(\mathbf{K}_{ss} - \mathbf{K}_f\right)\mathbf{u}_s = \mathbf{0} \tag{51}$$

and we solve it with Newton-Raphson. The residual Jacobian is

$$\frac{\partial \mathbf{r}(\mathbf{u}_s)}{\partial \mathbf{u}_s} = \mathbf{K}_{ss} - \mathbf{K}_f \tag{52}$$

3.1 Derivative computation

Now suppose that we want the derivative of several cost functions $\mathbf{f}(\mathbf{x}, \mathbf{u}_s)$, like lift or drag force. The total derivative as a form of partials [10, Sec. 6.7.2] is

$$\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}\mathbf{x}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \frac{\partial \mathbf{f}}{\partial \mathbf{u}_s}\frac{\mathrm{d}\mathbf{u}_s}{\mathrm{d}\mathbf{x}} \tag{53}$$

$$\boxed{\underbrace{\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}\mathbf{x}}}_{n_f \times n_x} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} - \underbrace{\frac{\partial \mathbf{f}}{\partial \mathbf{u}_s}\left[\frac{\partial \mathbf{r}}{\partial \mathbf{u}_s}\right]^{-1}}_{\boldsymbol{\Psi}^\top (n_f \times n_u)}\frac{\partial \mathbf{r}}{\partial \mathbf{x}}} \tag{54}$$

so to compute the total gradient, we can solve the *adjoint system*

$$\frac{\partial \mathbf{r}}{\partial \mathbf{u}_s}^\top \boldsymbol{\Psi} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}_s}^\top \qquad \text{where} \quad \boldsymbol{\Psi} = \begin{bmatrix} | & & | \\ \boldsymbol{\psi}_1 & \cdots & \boldsymbol{\psi}_{n_f} \\ | & & | \end{bmatrix} \tag{55}$$

so there are n_f right-hand sides in this equation. The partial derivative matrices in Equation (54) are computed through finite differences.

3.2 Elliptical lift

It is helpful to compare lift distributions ($L = \int_{-b/2}^{b/2} L'(y)dy$) to the elliptical lift distribution. An ellipse is

$$\gamma(y) = \Gamma_0\sqrt{1 - \left(\frac{2y}{b}\right)^2} \tag{56}$$

where the spanwise lift distribution from the Kutta-Joukowski theorem is $L'(y) = \rho U_\infty \gamma(y)$. Substituting the spanwise circulation into the Kutta-Joukowski theorem and the relation for total lift, we get

$$L = \int_{-b/2}^{b/2} L'(y)dy = \int_{-b/2}^{b/2} \rho U_\infty \Gamma_0 \sqrt{1 - \left(\frac{2y}{b}\right)^2}\, dy = \frac{\pi}{4}\rho U_\infty \Gamma_0 b \tag{57}$$

where the integral solution comes from integral tables or the geometric observation that the area of an ellipse is $\pi/4$ times the area of the bounding rectangle.

3.3 Total drag

Carlton [11, Fig. 21.3] and Chernyshev et al. [12, Fig. 4] provide references on total drag of ships. Raymer [13, Fig. 12.2] is also a good reference on drag classification. The ideas from both give us an idea of the calm-water drag of a foiling ship. In waves, an added resistance component due to radiated waves from the body responding to incident waves must be considered in total drag.

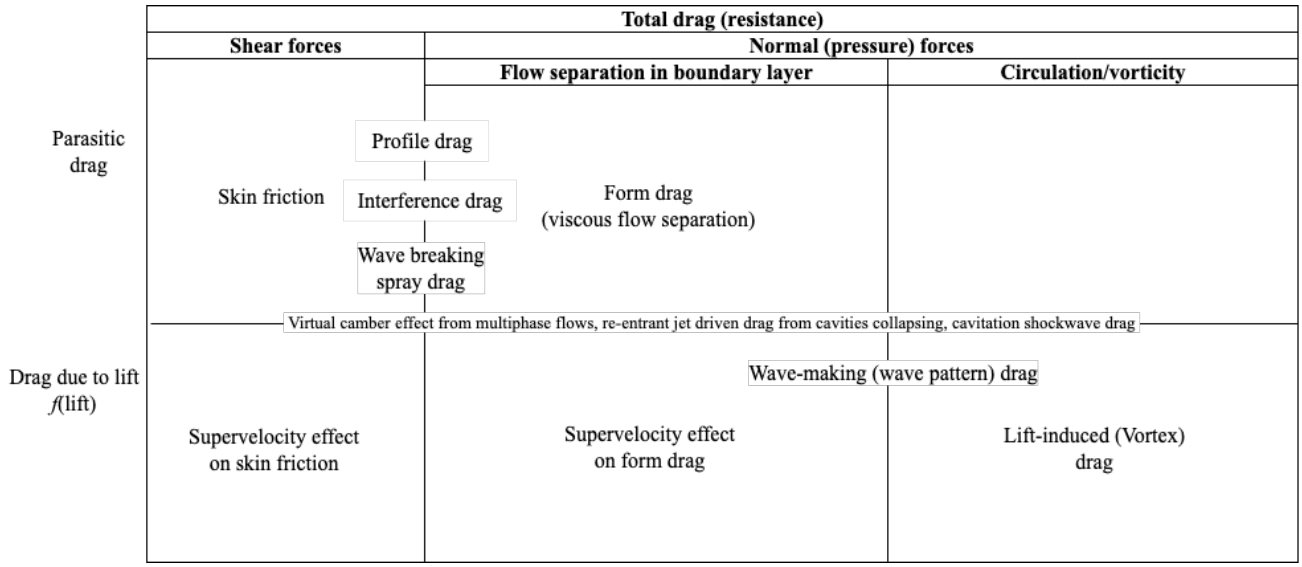

Figure 3: Hydrodynamic drag

4 Forced vibration solution

4.1 Governing equations

The frequency response solver separates the magnitudes of static and fluctuating parts of the solution $\mathbf{u_{tot}} = \mathbf{u_{dyn}} + \mathbf{u_{stat}}$. The static problem is solved the same way as previously described in, and we solve the dynamic part using the second-order dynamic governing equations for the user-prescribed harmonic forcing vector in the Laplace domain. We use $j = \sqrt{-1}$. Equation becomes

$$\underbrace{\left(-\omega^2 \left(\mathbf{M_f} + \mathbf{M_s}\right) + j\omega \left(\mathbf{C_f} + \mathbf{C_s}\right) + \left(\mathbf{K_f} + \mathbf{K_s}\right)\right)}_{\mathbf{D}(\omega)} \tilde{\mathbf{u}} = \tilde{\mathbf{f}}. \tag{58}$$

In this case, we substituted $\mathbf{u_{dyn}} = \tilde{\mathbf{u}} e^{j\omega t}$ and $\mathbf{f_{ext,dyn}} = \tilde{\mathbf{f}} e^{j\omega t}$ into since we are looking at forced harmonic vibration, and we care more about the steady-state (particular) solution than the initial transience (complementary solution). The system dynamic matrix ($\mathbf{D}(\omega)$), also known as the *impedance matrix*, is not symmetric because of the fluid governing equations. We solve for the dynamic response with direct inversion of the dynamic matrix. We solve this equation for a sweep of forcing frequencies ($f = \omega/(2\pi)$) and then compute the frequency response curves. The steady-state, frequency response is $\tilde{\mathbf{u}} = \mathbf{D}^{-1}\tilde{\mathbf{f}}$.

The inverse matrix $\mathbf{H}(\omega) = \mathbf{D}^{-1}(\omega)$ is the RAO or frequency response function (FRF), and it is an nDOF×nDOF matrix computed for every exciting frequency ω. Knowing the RAO is extremely important for understanding foil response to dynamic loading such as from waves or cavity shedding.

5 Flutter solution

5.1 Governing equations

The p-k method is commonly used in aeroelastic flutter predictions. The governing equation is solved by assuming a solution of the form $\mathbf{u} = \tilde{\mathbf{u}}e^{pt}$, where $p = \xi + jk$ is our non-dimensional complex eigenvalue, ξ is non-dimensional damping, and k is the reduced frequency. Eigenvalues are non-dimensionalized using $U_\infty \cos(\Lambda)/\bar{b}$ where $\bar{b}$ is the mean semi-chord. The generalized governing discrete equation takes the form

$$\left[\left(\frac{U_\infty \cos(\Lambda)}{\bar{b}}\right)^2 p_n^2 \left(\mathbf{M_s} + \mathbf{M_f}\right) + \frac{U_\infty \cos(\Lambda)}{\bar{b}} p_n \mathbf{C_s} + \mathbf{K_s} - \mathbf{f_{hydro,quasi}}\right] \tilde{\mathbf{u}}_n = \mathbf{0} \qquad \text{for } n = 1, \ldots, \text{nmode}. \quad (59)$$

where $\mathbf{f_{hydro,quasi}}$ are the quasi-steady hydrodynamic forces. For a non-trivial solution of $\tilde{\mathbf{u}}$, the flutter determinant $\det(\Delta)$ is zero, where the bracketed matrix in Equation (59) is Δ. We need to find the p's that satisfies the quadratic eigenproblem so that $\det(\Delta) = 0$. The corresponding $\tilde{\mathbf{u}}$ is then the complex eigenvector describing the flutter mode shape. Simulations sweep flow speed (U_∞). We can compute in-vacuum natural frequencies by setting all fluid matrices and total damping to zero, giving a linear eigenvalue problem with symmetric matrices; the eigenvector is the corresponding in-vacuum mode shape.

We can linearize the quadratic eigenvalue problem using the identity $\mathbf{I}\dot{\mathbf{u}} = \mathbf{I}\dot{\mathbf{u}}$ to obtain this linear, generalized eigenvalue problem

$$p_n \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \left(\frac{U_\infty \cos(\Lambda)}{\bar{b}}\right)^2 \mathbf{M} \end{bmatrix} \begin{Bmatrix} \tilde{\mathbf{u}} \\ p_n \tilde{\mathbf{u}} \end{Bmatrix} - \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K} & -\frac{U_\infty \cos(\Lambda)}{\bar{b}} \mathbf{C} \end{bmatrix} \begin{Bmatrix} \tilde{\mathbf{u}} \\ p_n \tilde{\mathbf{u}} \end{Bmatrix} = \mathbf{0}, \quad (60)$$

We solve Equation (60) following the method of van Zyl [14] since Jonsson [15] has shown the method is suitable and sufficiently robust enough for gradient-based optimization. The flutter solution method is

1. For a given velocity, solve the eigenvalue problem (Equation (60)) for a range of reduced frequencies k.
2. Eigenvalues (p) are valid roots if $\Im(p)$ matches the assumed k (matched point solution).
3. Monitor the difference $\Im(p) - k$. A change in sign indicates a root.
4. Determine the root by linear interpolation.

where we follow the methodology outlined in Jonsson et al. [16] and Jonsson [15, Ch. 3]. Since the linear system is quite large, we also use a mode space reduction technique commonly known as the *normal mode* (or modal) method to reduce the size of the matrices and computational cost.

5.2 Mode tracking

The mode tracking method prevents mode swapping or hopping during two stages of the flutter analysis:

1. during the reduced frequency sweep at a given flight condition and
2. during the dynamic pressure increments between flight conditions.

We employ the mode tracking method from van Zyl [17], which uses complex inner products between current and previous eigenvectors to populate a correlation matrix. $\tilde{\mathbf{C}}$. We search the matrix for largest elements (maximum of one) where the row and column denote previous iteration and current iteration, respectively. We do this for all modes. In the frequency sweep, building $\tilde{\mathbf{C}}$ is simpler because it is square as the number of eigenvectors between k iterations is the same, resulting in one-to-one mapping. Between dynamic pressure increments, modes can show up or disappear so $\tilde{\mathbf{C}}$ is rectangular. Therefore, we use a correlation metric (vector $\mathbf{c}$) to determine if the computed set of eigenvalues are too far from previously computed values, the process of which is outlined in Algorithm 2. Based on the correlation metric value, we either accept or reject the eigenvalues and eigenvectors. If they are rejected, the dynamic pressure step is halved and we re-run the reduced frequency sweep; however, the halving process is controlled by a minimum allowed increment $\Delta q_{\min}$ to prevent excessively long computation times. If $\Delta q_{\min}$ is reached, then we accept the roots.

Algorithm 2: Mode tracking between dynamic pressure increments $q^{(n)}$ and $q^{(n+1)}$ of Jonsson [15] adapted from van Zyl [17].

Data: Complex eigenvector matrix $\mathbf{V}^{(n)}$ and $\mathbf{V}^{(n+1)}$ at succesive q's of size $n_{\text{dof}} \times n_v^{(n)}$ and $n_{\text{dof}} \times n_v^{(n+1)}$
Result: Vector $\mathbf{c}$ of size $n_v^{(n+1)}$ and array $\mathbf{m}$ of size $2 \times n_v^{(n+1)}$

1 Compute $S_{3i} = ||\mathbf{v}_i^{(n)}||_2$ and $S_{4i} = ||\mathbf{v}_i^{(n+1)}||_2$;
2 Compute Hadamard product $\mathbf{C} = |\mathbf{V}^{(n)^H} \circ \mathbf{V}^{(n+1)}|$;
3 $\tilde{\mathbf{C}} = \mathbf{C}_{ij} / \left(S_{3i} S_{4j}\right)$;
4 Search $\tilde{\mathbf{C}}$ for maximum element;
5 Store row and column indices in array $\mathbf{m}$ and then the correlation values in $\mathbf{c}$;
6 Set the corresponding rows and columns in $\tilde{\mathbf{C}}$ to zero;
7 Repeat process until all elements in $\tilde{\mathbf{C}}$ are zero;

5.3 Mode space reduction

To reduce the problem size, we use mode space reduction to a reduced set of N_r generalized coordinates. The displacement field is approximated by

$$\mathbf{u} \approx \mathbf{Q}_r(y)\mathbf{q}(t) \tag{61}$$

where $\mathbf{q} \in \mathbb{R}^{N_r}$ is a vector of retained generalized coordinates and $\mathbf{Q}_r \in \mathbb{R}^{N_s \times N_r}$ is a matrix with columns corresponding to eigenvectors. This is typically called the normal mode method. One then solves the eigenvalue problem

$$\left(\mathbf{K}_\mathbf{s} - \omega_i^2 \mathbf{M}_\mathbf{s}\right) \bar{\mathbf{u}}_i = 0 \tag{62}$$

where ω_i is the natural frequency. We compute $\bar{\mathbf{u}}_i$ and collect them in the matrix

$$\mathbf{Q}_r = \begin{bmatrix} | & | & & | \\ \bar{\mathbf{u}}_1 & \bar{\mathbf{u}}_2 & \cdots & \bar{\mathbf{u}}_{N_r} \\ | & | & & | \end{bmatrix}. \tag{63}$$

Now the reduced stiffness and mass matrices are

$$\mathbf{M}_{\mathbf{s}r} = \mathbf{Q}_r^\top \mathbf{M}_\mathbf{s} \mathbf{Q}_r = \mathbf{I}_r \in \mathbb{R}^{N_r \times N_r} \tag{64}$$

$$\mathbf{K}_{\mathbf{s}r} = \mathbf{Q}_r^\top \mathbf{K}_\mathbf{s} \mathbf{Q}_r = \text{diag}\left[\omega_i^2\right] \tag{65}$$

and the governing equation reduces to

$$\mathbf{M}_{\mathbf{s}r}\ddot{\mathbf{q}} + \mathbf{C}_{\mathbf{s}r}\ddot{\mathbf{q}} + \mathbf{K}_{\mathbf{s}r}\mathbf{q} - \mathbf{Q}_r^\top \mathbf{f}_{\text{hydro}} = \mathbf{0} \tag{66}$$

To apply this to the hydrodynamic loads, we obtain from Equation (**??**)

$$\mathbf{f}_{\text{hydro},r} = -\left(\mathbf{M}_{\mathbf{f}r}\ddot{\mathbf{u}} + \mathbf{C}_{\mathbf{f}r}\dot{\mathbf{u}} + \mathbf{K}_{\mathbf{f}r}\mathbf{u}\right) \tag{67}$$

where the matrices are

$$\mathbf{M}_{\mathbf{f}r} = \mathbf{Q}_r^\top \mathbf{M}_\mathbf{f} \mathbf{Q}_r \tag{68}$$

$$\mathbf{C}_{\mathbf{f}r} = \mathbf{Q}_r^\top \mathbf{C}_\mathbf{f} \mathbf{Q}_r \tag{69}$$

$$\mathbf{K}_{\mathbf{f}r} = \mathbf{Q}_r^\top \mathbf{K}_\mathbf{f} \mathbf{Q}_r. \tag{70}$$

Note, since cavitating flow leads to non-symmetric matrices, we cannot do all the simplifications Jonsson et al. [18] uses. TODO: OK, this is not actually true because we wouldn't be able to solve in the frequency domain...

5.4 Cost functions

Reverse mode algorithmic differentiation is applied on the upper-most flutter routine called `cost_funcs_with_derivs`

5.4.1 Flutter

The single flutter constraint uses the damping values ($g = \Re\{p\}$). We use a two-level KS function aggregation to compute the single constraint first used by Jonsson et al. [18]. It is

$$\mathrm{KS}_{\text{flutter}} = \mathrm{KS}\left(\mathrm{KS}\left(\Re\{p_{n,q}\}\right)\right) \tag{71}$$

where the first level is over all dynamic pressures $q = 1, \ldots, N_q$ and then over all modes $n = 1, \ldots, \text{nmodes}$. The KS function is

$$\mathrm{KS}(\mathbf{a}) = a_{\max} + \frac{1}{\rho} \ln \sum_i e^{\rho(a_i - a_{\max})} \tag{72}$$

where we set the aggregation parameter $\rho = 80$ by default.

5.4.2 Coalescence

6 Rigid body dynamics

The two coordinate systems are the *earth-fixed* and *body-fixed* (stability) axes, abbreviated EFS and BFS. DCFoil assumes an equilibrium state is provided (such as from a VPP).

6.1 Governing equations

The BFS and EFS origin is at the center of mass of the craft. Forward is x, port is y, and up is z. These are different axes from the foil. There are six degrees of freedom in the BFS denoted $\boldsymbol{\eta} = [\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6]^\top$. The six velocities in BFS are $[v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]^\top$ where the first three are $\mathbf{v}_B$ and the last three are $\boldsymbol{\omega}_B$. The orientation in EFS are *Euler angles* $[\phi, \theta, \psi]$. See Figure 4.

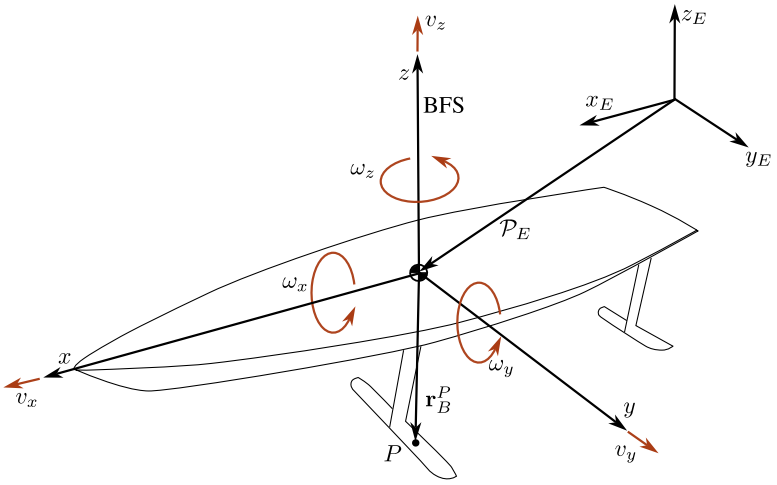

Figure 4: Coordinate system and degrees of freedom

6.2 Kinematics

A quantity of interest is the vessel velocity in the EFS denoted $\mathbf{v}_E$. Using Euler angles (from EFS obviously), the relationship between BFS and EFS center of mass velocity is

$$\mathbf{v}_E = \mathbf{R}_{BE}\mathbf{v}_B.$$

Now for some point P on the vessel not at the center of mass, the velocity is

$$\mathbf{v}_E^P = \dot{\mathcal{P}}_E + \dot{\mathbf{r}}_E^P = \mathbf{v}_E + \underbrace{\dot{\mathbf{r}}_E^P}_{\dot{\mathbf{R}}_{BE}\mathbf{r}_B^P},$$

where ship velocity is $\mathbf{v}_E = \dot{\mathcal{P}}_E$. The orientation of the ship in EFS is

$$\boxed{\begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix} = \mathbf{T}^{-1}\boldsymbol{\omega}_B = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix}}. \tag{73}$$

This is the equation you would plug into the time integration scheme. The *tangential operator* $\mathbf{T}(\phi, \theta, \psi)$ based on Euler angles is

$$\mathbf{T}(\phi, \theta, \psi) = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix}.$$

We wrote it this way to not use Euler angles. Finally, the *transport theorem* is

$$\mathbf{R}_{BE}\dot{\mathbf{v}}_E = \dot{\mathbf{v}}_B + \tilde{\boldsymbol{\omega}}_B\mathbf{v}_B$$

where

$$\mathbf{R}_{BE}\dot{\mathbf{R}}_{EB} = \tilde{\boldsymbol{\omega}}_B = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}.$$

6.3 Inertia

Total linear momentum is

$$\mathbf{p}_B = \int_{\mathcal{V}} \mathbf{v}_B^P \, \mathrm{d}m = m\mathbf{v}_B$$

and the total angular momentum about the center of mass is

$$\mathbf{h}_B = \int_{\mathcal{V}} \tilde{\mathbf{r}}_B^P \mathbf{v}_B^P \, \mathrm{d}m = \mathbf{I}_B\boldsymbol{\omega}_B$$

where

$$\mathbf{I}_B = -\int_{\mathcal{V}} \tilde{\mathbf{r}}_B^P \tilde{\mathbf{r}}_B^P \, \mathrm{d}m = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix}$$

and $I_{xy} = I_{zy} = 0$ for P/S symmetry. Example quantities are $I_{xx} = \int_{\mathcal{V}} \left(y^2 + z^2\right) dm$ and $I_{yz} = I_{zy} = \int_{\mathcal{V}} yz dm$ In EFS, angular momentum is

$$\mathbf{h}_E = \mathbf{R}_{EB}\mathbf{h}_B = \underbrace{\mathbf{R}_{EB}\mathbf{I}_B\mathbf{R}_{BE}}_{\mathbf{I}_E}\boldsymbol{\omega}_E$$

6.4 Equations of motion

If we plug into Lagrange's equations, we get the equations of motion in BFS

$$m\dot{\mathbf{v}}_B + m\tilde{\boldsymbol{\omega}}_B\mathbf{v}_B = \mathbf{f}_{\mathrm{h}B} + \mathbf{f}_{pB} + \mathbf{f}_{\mathrm{g}B} \tag{74}$$

$$\mathbf{I}_B\dot{\boldsymbol{\omega}}_B + \tilde{\boldsymbol{\omega}}_B\mathbf{I}_B\boldsymbol{\omega}_B = \mathbf{m}_{aB} + \mathbf{m}_{pB} \tag{75}$$

where the RHS are hydrodynamic, propulsive, and gravitational loads acting on the center of mass.

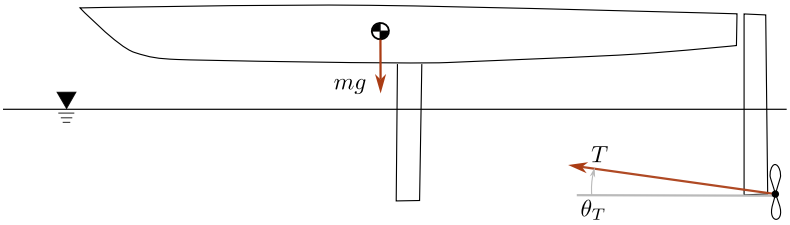

Figure 5: Gravitational and propulsive loads.

6.4.1 Gravitational loads

The gravitational loads have explicit dependence on vehicle orientation.

$$\mathbf{f}_{gB} = mg\mathbf{R}_{BE}\mathbf{e}_3 = \left\{ \begin{array}{c} -mg\sin\theta \\ mg\cos\theta\sin\phi \\ mg\cos\theta\cos\phi \end{array} \right\}$$

6.4.2 Propulsive forces

Assume the propulsion system acts at a point at $(x_T, 0, z_T)$ from the CM with some rake angle θ_T. The propulsive loads are then

$$\mathbf{f}_{pB} = \left\{ \begin{array}{c} T\cos\theta_T \\ 0 \\ -T\sin\theta_T \end{array} \right\}, \quad \mathbf{m}_{pB} = \left\{ \begin{array}{c} 0 \\ T\left(z_T\cos\theta_T + x_T\sin\theta_T\right) \\ 0 \end{array} \right\}$$

6.4.3 Hydrodynamic loads

The craft instantaneous angle of attack is

$$\alpha(t) = \tan^{-1}\frac{v_z(t) - v_{gz}(t)}{v_x(t) - v_{gx}(t)} \quad \text{and} \quad -\pi \le \alpha \le \pi$$

where $\mathbf{w} = -[v_{gx}, v_{gy}, v_{gz}, \omega_{gx}, \omega_{gy}, \omega_{gz}]^\top$ is the disturbance vector of gusts (assume constant). Leeway angle is

$$\beta = \tan^{-1}\frac{v_y - v_{gy}}{\sqrt{\left(v_x - v_{gx}\right)^2 + \left(v_z - v_{gz}\right)^2}} \quad \text{and} \quad -\frac{\pi}{2} \le \beta \le \frac{\pi}{2}.$$

Finally, the velocity vector is

$$\mathbf{v}_B - \mathbf{v}_{gB} = \left\{ \begin{array}{c} v_x - v_{gx} \\ v_y - v_{gy} \\ v_z - v_{gz} \end{array} \right\} = V_{\text{TAS}} \left\{ \begin{array}{c} \cos\alpha\cos\beta \\ -\sin\beta \\ \sin\alpha\cos\beta \end{array} \right\}$$

where $V_{TAS} = ||\mathbf{v}_B - \mathbf{v}_{gB}||$ is the magnitude of the true foiling speed. The hydrodynamics loads in body axes are

$$\begin{aligned} \mathbf{f}_{hB} &= \frac{1}{2}\rho V_{\text{TAS}}^2 S\mathbf{c}_{\mathbf{f}_B}\left(\mathbf{v}_B - \mathbf{v}_{gB}, \boldsymbol{\omega}_B - \boldsymbol{\omega}_{gB}, \boldsymbol{\delta}_c; \sigma, \text{Re}\right), \\ \mathbf{m}_{hB} &= \frac{1}{2}\rho V_{\text{TAS}}^2 S\boldsymbol{\Lambda}_{\text{ref}}\mathbf{c}_{\mathbf{m}_B}\left(\mathbf{v}_B - \mathbf{v}_{gB}, \boldsymbol{\omega}_B - \boldsymbol{\omega}_{gB}, \boldsymbol{\delta}_c; \sigma, \text{Re}\right), \end{aligned}$$

where $\mathbf{c}_{\mathbf{f}_B}$ and $\mathbf{c}_{\mathbf{m}_B}$ are the force and moment coefficients, respectively. The vector $\boldsymbol{\delta}_c$ is the vector of angles of the control surfaces.

The hydrodynamic stability derivatives allow us to tune $\boldsymbol{\delta}_c$ to find incremental changes in the hydrodynamic loads.

$$\Delta\mathbf{f}_{hB} = qS\left(\Delta\mathbf{c}_{\mathbf{f}_B}+\right) \tag{76}$$

$$\Delta\mathbf{m}_{hB} = qS\left(\Delta\mathbf{c}_{\mathbf{m}_B}+\right) \tag{77}$$

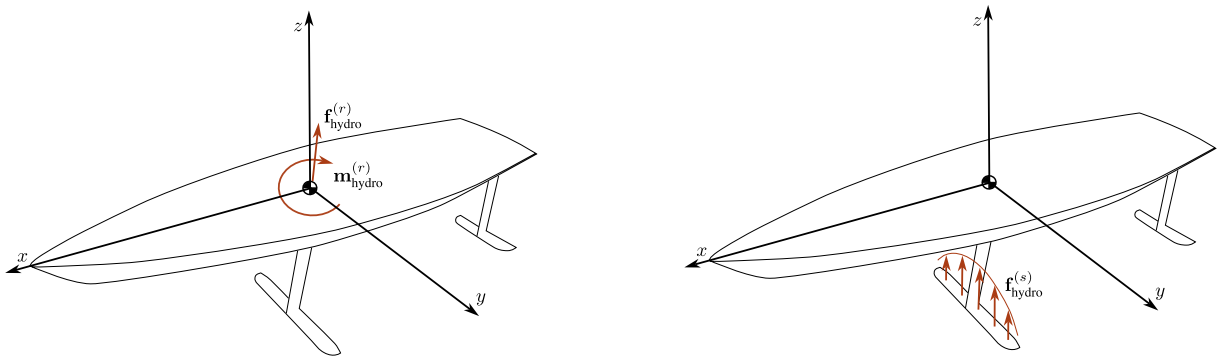

Figure 6: Hydrodynamic loads.

6.5 Steady-state flight solution

In steady-state flight, there is no acceleration so $\dot{\mathbf{v}}_B = \dot{\boldsymbol{\omega}}_B = 0$. Furthermore, $\dot{\phi} = \dot{\theta} = \dot{\psi} = 0, \phi = \theta = 0$.

$$\boxed{\mathbf{f}_{hB} + \mathbf{f}_{pB} + \mathbf{f}_{gB} - m\tilde{\boldsymbol{\omega}}_B \mathbf{v}_B = \mathbf{0}} \tag{78}$$

$$\boxed{\mathbf{m}_{hB} + \mathbf{m}_{pB} - \tilde{\boldsymbol{\omega}}_B \mathbf{I}_B \boldsymbol{\omega}_B = \mathbf{0}} \tag{79}$$

The pitch of the vessel θ is treated differently from the trim of the boat τ. The problem is now finding the $\boldsymbol{\delta}_c$ that satisfy the above equations where the control angles are vessel trim and rudder rake.

$$\boldsymbol{\delta}_c = [\tau \quad \delta_r]^\top \tag{80}$$

The problem is then solving for the $\boldsymbol{\delta_c}$ that satisfies Equation (78). This can be done with a simple Newton-Raphson root-finding method.

The residual Jacobian needed is

(81)

6.5.1 Elastified equations of motion

Defining the state vector as $\mathbf{x}_r(t) = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z, \phi, \theta]^\top$ and the control inputs as $\mathbf{u} = [\tau, \delta_r]^\top$ where τ and δ_r are vessel trim and rudder rake command inputs, the dynamics of the system are

$$\dot{\mathbf{x}}_r = f^{(r)}(\mathbf{x}_r, \mathbf{x}_s, \mathbf{u}, \mathbf{w}) = \tag{82}$$

$$f^{(r)}_{\text{gyr}}(\mathbf{x}_r) + f_{\text{grav}}(\mathbf{x}_r) + f^{(r)}_{\text{hydro}}(\mathbf{x}_r, \mathbf{x}_s, \mathbf{u}) + f^{(r)}_{\text{prop}}(\mathbf{x}_r, \mathbf{x}_s, \mathbf{u}) \tag{83}$$

and (under the small amplitude structural deformation assumption) the deformations are determined by

$$\mathbf{K}_{ss}\mathbf{x}_s = f^{(s)}(\mathbf{x}_r, \mathbf{x_s}, \mathbf{u}) = \tag{84}$$

$$f^{(s)}_{\text{gyr}}(\mathbf{x}_r) + f_{\text{grav}}(\mathbf{x}_r) + f^{(s)}_{\text{hydro}}(\mathbf{x}_r, \mathbf{x}_s, \mathbf{u}) + f^{(s)}_{\text{prop}}(\mathbf{x}_r, \mathbf{x}_s, \mathbf{u}) \tag{85}$$

where $\mathbf{x}_s(t)$ is the structural state vector.

Equations (82) and (84) are coupled and must be solved simultaneously. Static trim is $\mathbf{x}_{s0}$ and $\mathbf{x}_{r0}$. We solve the coupled system

$$f^{(r)}(\mathbf{x}_{s0}, \mathbf{x}_{r0}, \mathbf{u}_0) = \mathbf{0} \tag{86}$$

$$f^{(s)}(\mathbf{x}_{s0}, \mathbf{x}_{r0}, \mathbf{u}_0) = \mathbf{K}_{ss}\mathbf{x}_{s0} \tag{87}$$

Assuming small perturbations around the trimmed flight condition, the force Equations (82) and (84) can be linearized to

$$\Delta\dot{\mathbf{x}}_r = \mathbf{A}_{rr}\Delta\mathbf{x}_r + \mathbf{A}_{rs}\Delta\mathbf{x}_s + \mathbf{B}_r\Delta\mathbf{u}$$

$$\mathbf{K}_{ss}\Delta\mathbf{x}_s = \mathbf{A}_{sr}\Delta\mathbf{x}_r + \mathbf{A}_{ss}\Delta\mathbf{x}_s + \mathbf{B}_s\Delta\mathbf{u}$$

where the state (**A**) and input (**B**) matrices are partitions of the Jacobian of nonlinear forcing terms. This coupled system is linear so we can first solve for

$$\Delta \mathbf{x}_s = \left(\mathbf{K}_{ss} - \mathbf{A}_{ss}\right)^{-1} \left(\mathbf{A}_{sr}\Delta \mathbf{x}_r + \mathbf{B}_s \Delta \mathbf{u}\right)$$

and this is substituted back into the linearized coupled equations. The structural effects are a constant feedback on the rigid equations.

Now in the vibrating mode, if we assume small-amplitude vibrations, we must consider the shift of vehicle center of mass and its impact on the vehicle dynamics. If the elastic deformations are sufficiently small to not change global inertial characteristics, then modal coordinates can be used. We introduce a *body-attached frame* (BAF) rigidly linked to a point on the vessel and a *floating frame* (FF).

7 Data structure and problem setup

The overall data flow in the program is in Figure 7. Appendage setup is through dictionaries.

8 Writing derivative routines

The forward and reverse mode AD logic from Martins and Ning [10] is

$$\frac{\mathrm{d}v_i}{\mathrm{d}v_j} = \dot{v}_i = \sum_{k=j}^{i-1} \frac{\partial v_i}{\partial v_k} \underbrace{\dot{v}_k}_{\frac{\mathrm{d}v_k}{\mathrm{d}v_j}}$$

$$\frac{\mathrm{d}v_i}{\mathrm{d}v_j} = \bar{v}_j = \sum_{k=j+1}^{i} \frac{\partial v_k}{\partial v_j} \underbrace{\bar{v}_k}_{\frac{\mathrm{d}v_i}{\mathrm{d}v_k}}$$

where the goal is the LHS (alternative notation $\frac{\mathrm{d}y}{\mathrm{d}x}$) over a massive length of code.

The reverse AD routines are in Julia syntax. Consider the primal function somewhere in the chain of code

$$y = f(x)$$

with a Jacobian matrix dy/dx. The vector-Jacobian product computed by $\mathcal{B}$ is

$$\frac{\partial a}{\partial x} = \underbrace{\frac{\partial a}{\partial y}}_{\bar{y}} \underbrace{\frac{\mathrm{d}y}{\mathrm{d}x}}_{\text{Jacobian}} = \mathcal{B}(\bar{y})$$

where $\bar{y}$ is called the *cotangent* vector and a is some known quantity at this stage of the chain rule. The analog in forward mode are the tangent vector and pushforward function.

The `pullback` functionality in AD packages typically returns two things: the evaluated point $f(x)$ (also called primal) and the pullback function $\mathcal{B}(\bar{y})$ There are situations in which one has a manual implementation they want the AD package to bypass, for example, matrix operations.

We write manual rules through `ChainRules.jl`. An example is Algorithm 3.

Algorithm 3: Basic reverse rule in Julia ChainRulesCore

function ChainRulesCore.rrule(::typeof($f(x)$), x)
  Evaluate $y = f(x)$ (Primal)
  function $\mathcal{B}(\bar{y})$
    Code up $\bar{x} = \bar{y}\frac{\mathrm{d}y}{\mathrm{d}x}$ (Ex. $\cos(x)\bar{y}$ if $f(x) = \sin(x)$)
  return $\bar{x}$
return y, $\mathcal{B}()$

Now that you understand the conceptual level, this is a more applied example for the eigenvalue derivative routine.

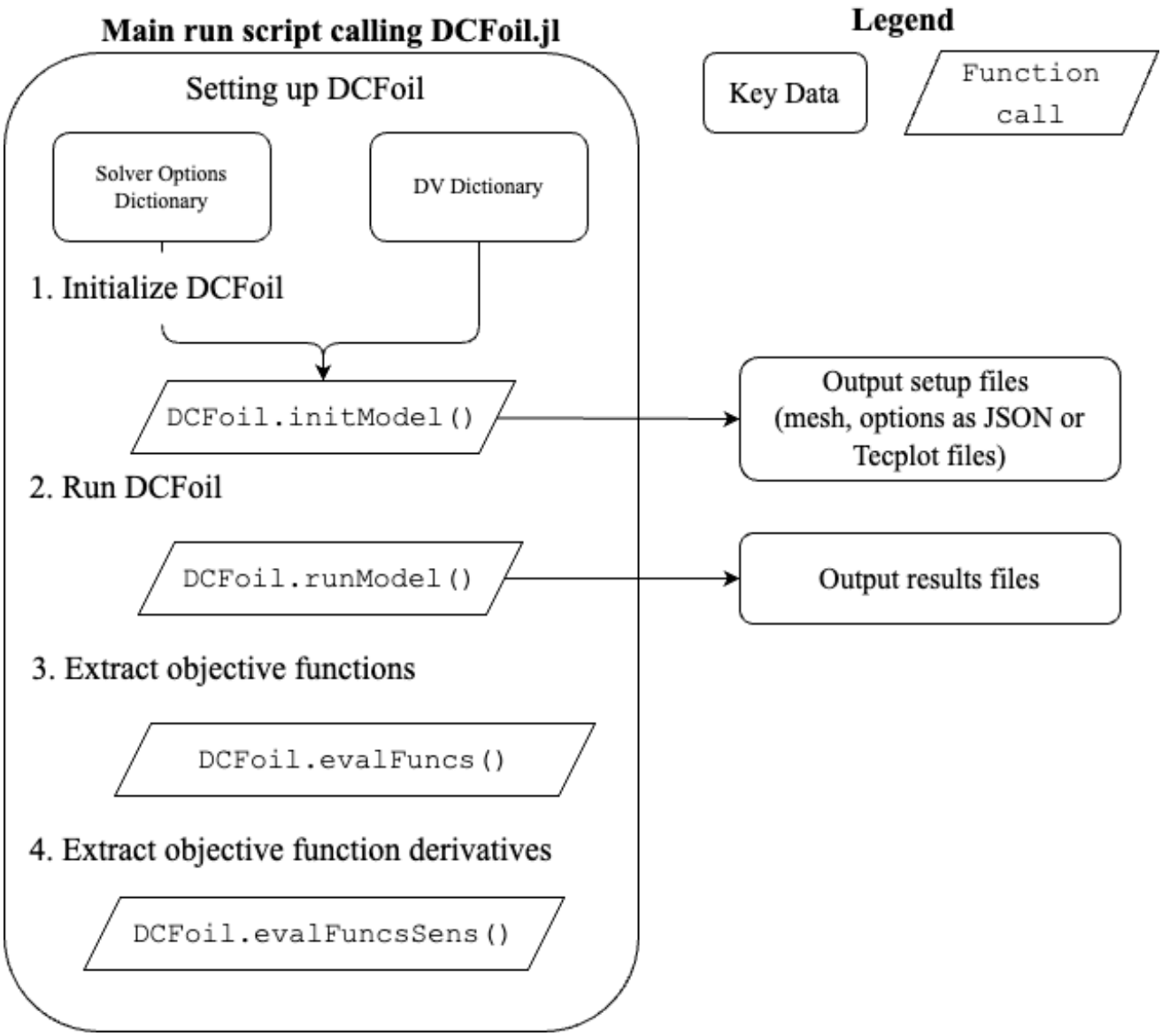

Figure 7: Data flow in the program.

Algorithm 4: Reverse rule for eigenvalue problem

function ChainRulesCore.rrule(::typeof(eigvals()), A::AbstractMatrix)

$\lambda_n, \phi_n = \text{eigvals}(A)$

return

References

[1] Terrence A Weisshaar and Brian L Foist. Vibration tailoring of advanced composite lifting surfaces. *Journal of Aircraft*, 22(2):141–147, 1985.

[2] I. Lottati. Flutter and divergence aeroelastic characteristics for composite forward swept cantilevered wing. *Journal of Aircraft*, 22(11):1001–1007, nov 1985. doi: 10.2514/3.45238.

[3] Hermann Glauert. *The Elements of Aerofoil and Airscrew Theory*. Cambridge Science Classics. Cambridge University Press, 1983. doi: 10.1017/CBO9780511574481.

[4] Justin E Kerwin and Jacques B Hadler. Principles of naval architecture series: Propulsion. *The Society of Naval Architects and Marine Engineers (SNAME)*, pages 18–30, 2010.

[5] Hermann Glauert. A theory of thin aerofoils. Technical report, 1924.

[6] Hermann Glauert. Theoretical relationships for an aerofoil with hinged flap. Technical report, Aeronautical research committee reports and memoranda, 1927.

[7] T. Theodorsen. General theory of aerodynamic instability and the mechanism of flutter. Technical Report Rept. 496, NACA, May 1934.

[8] Yusong Cao, William W Schultz, and Robert F Beck. Three-dimensional desingularized boundary integral methods for potential problems. *International Journal for Numerical Methods in Fluids*, 12(8):785–803, 1991.

[9] J. L. Hess and A. M. O. Smith. Calculation of potential flow about arbitrary bodies. *Progress in Aerospace Sciences*, 8(1–138), 1967.

[10] Joaquim R. R. A. Martins and Andrew Ning. *Engineering Design Optimization*. Cambridge University Press, Cambridge, UK, January 2022. ISBN 9781108833417. doi: 10.1017/9781108980647. URL `https://mdobook.github.io`.

[11] John Carlton. *Marine Propellers and Propulsion*. Butterworth-Heinemann, 2018.

[12] Lev Chernyshev, Natalia Kabaliuk, Mark Jermy, Simon Corkery, and Daniel Bernasconi. Determining the physical components of resistance acting on a hydrofoil. volume Volume 9: Fluids Engineering of *ASME International Mechanical Engineering Congress and Exposition*, October 2023. doi: 10.1115/IMECE2023-112475.

[13] Daniel P. Raymer. *Aircraft Design: A Conceptual Approach*. AIAA, Reston, VA, 5th edition, 2012.

[14] Louw H. van Zyl. Aeroelastic divergence and aerodynamic lag roots. *Journal of Aircraft*, 38(3):586–588, May 2001. ISSN 0021-8669. doi: 10.2514/2.2806. URL `http://dx.doi.org/10.2514/2.2806`.

[15] Eirikur Jonsson. *High-fidelity Aerostructural Optimization of Flexible Wings with Flutter Constraints*. PhD thesis, University of Michigan, 2020.

[16] Eirikur Jonsson, Charles A. Mader, Graeme J. Kennedy, and Joaquim R. R. A. Martins. Computational modeling of flutter constraint for high-fidelity aerostructural optimization. In *2019 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, San Diego, CA, January 2019. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2019-2354.

[17] Louw H. van Zyl. Use of eigenvectors in the solution of the flutter equation. *Journal of Aircraft*, 30(4):553–554, July 1993. ISSN 0021-8669. doi: 10.2514/3.46380. URL `http://dx.doi.org/10.2514/3.46380`.

[18] Eirikur Jonsson, Gaetan K. W. Kenway, Graeme J. Kennedy, and Joaquim R. R. A. Martins. Development of flutter constraints for high-fidelity aerostructural optimization. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Denver, CO, June 2017. AIAA 2017-4455.