



## INTERIM REPORT 2

AERO321

DYNAMICS OF AEROSPACE VEHICLES

---

## Fall 23 Project

---

### *Group Members*

Benjamin Tollison

Jirina Bredberg

Ines Meyer

Kanishka Kamal

### *Tasks*

Author

Tables 1,2

Tables 1,3

Tables 3,4

Wednesday 22<sup>nd</sup> November, 2023

# Contents

<b>1</b>	<b>Task 1a</b>	<b>3</b>
1.1	Determining $L_{WF}$	3
1.2	Finding $C_{L_{\alpha, W}}$	3
1.3	Find $q_{\infty}$	3
1.4	Finding $D_0$	4
1.5	Finding $L_H$	4
1.6	Finding $F_{N_0}$	4
1.7	Finding $\epsilon$	5
<b>2</b>	<b>Task 1b</b>	<b>5</b>
2.1	Equations to consider	5
2.2	Algorithm	6
<b>3</b>	<b>Task 1 Tables</b>	<b>7</b>
<b>4</b>	<b>Task 2</b>	<b>10</b>
4.1	Analysis of the Longitudinal Derivatives	12
4.2	Explanation of Dynamic Parameters from Eigenvalues	12
4.3	Categorizing the Aircraft	13
4.4	Task 2 Tables	13

# 1 Task 1a

Determine the trim AoA, tail incidence angle, equilibrium thrust, and the normal force with the following system of equations in the body-axis system:

$$\begin{aligned}\sum F_X = 0 &= T_0 - W \sin \alpha_0 + L_{WF} \sin \alpha_0 - D_0 \cos \alpha_0 + L_H \sin(\alpha_0 - \epsilon) \\ \sum F_Z = 0 &= -F_{N_0} + W \cos \alpha_0 - L_{WF} \cos \alpha_0 - D_0 \sin \alpha_0 - L_H \cos(\alpha_0 - \epsilon) \\ \sum M_{cg} = 0 &= F_{N_0}(X_{cg} - X_{inlet}) - L_{WF}(X_{AC_{WF}} - X_{cg}) \cos \alpha_0 \\ &\quad - D_0(X_{AC_{WF}} - X_{cg}) \sin \alpha_0 - L_H(X_{AC_H} - X_{cg}) \cos(\alpha_0 - \epsilon)\end{aligned}$$

A systematic approach to solving a system of equations involves breaking down the problem into manageable steps. Initially, one focuses on solving each equation by isolating the capitalized coefficients. This often requires employing linear approximations to better understand the behavior of each equation. The next step is to deconstruct the equations further, systematically breaking them down into simpler forms that are easier to solve. This iterative process allows for a methodical exploration of the system's intricacies, gradually unveiling solutions and facilitating a more comprehensive understanding of the relationships among the variables. Through this systematic method, one navigates the complexity of the system, step by step, until arriving at a point where the equations can be effectively solved.

## 1.1 Determining $L_{WF}$

It is stated that we can neglect the lift from the fuselage in the project assignment such that

$$L_{WF} \approx L_W = q_\infty S_W C_{L_W}$$

Find  $C_{L_W} \approx C_{L_{W_0}} + C_{L_{\alpha, W}} \alpha_0$

where  $C_{L_{W_0}} = 0$  due to symmetry, which leads to the problem that we need to now find  $C_{L_{\alpha, W}}$

## 1.2 Finding $C_{L_{\alpha, W}}$

1. Find Taper ratio  $\lambda = \frac{2S_w}{bc_r}$
2. Find SweepAngle of the LE:  $\Lambda_{LE} = \tan^{-1} \left( \tan \Lambda_{c/4} + \frac{1}{AR} \frac{1-\lambda}{1+\lambda} \right)$
3. Find SweepAngle of the c/2:  $\Lambda_{c/2} = \tan^{-1} \left( \tan \Lambda_{LE} - \frac{2}{AR} \frac{1-\lambda}{1+\lambda} \right)$
4. Find  $C_{l_\alpha} = \frac{2\pi}{\sqrt{1-M_\infty^2}}$
5. Find beta/k constants:  $\beta = \sqrt{1-M^2}$ ,  $k = \frac{C_{l_\alpha}}{2\pi}$
6. plug into equation:

$$C_{L_\alpha} = \frac{2\pi AR}{2 + \sqrt{\frac{AR^2 \beta^2}{k^2} \left( 1 + \frac{\tan^2 \Lambda_{c/2}}{\beta^2} \right)} + 4}$$

## 1.3 Find $q_\infty$

$$q_\infty = \frac{1}{2} \rho_\infty v_\infty^2$$

1. Get speed of sound from the tables:  $a = 296.5338[m/s]$

2. Find  $v_\infty$  from  $M = \frac{v_\infty}{a}$

3. Get the  $\rho_\infty$  with isentropic equation

$$(a) \frac{\rho_0}{\rho} = \left(1 + \frac{\gamma-1}{2} M^2\right)^{\frac{1}{\gamma-1}}$$

$$(b) \rho_0 = 7.3654 \cdot 10^{-4} [sl/ft^3] = 0.3795971165326 [kg/m^3]$$

$$\therefore L_{WF} = q_\infty S_W C_{L_{\alpha, W}} \alpha_0$$

## 1.4 Finding $D_0$

$$\rightarrow D_0 \equiv D(\alpha_0)$$

$$D = q_\infty S_W C_D$$

1. Find  $C_D$  from the given equation:  $C_D = 0.0145 + 0.1 C_{L_W}^2$

$$(a) \text{ with } C_{L_W} = C_{L_{\alpha, W}} \alpha_0$$

$$\therefore D_0 = q_\infty S_W (0.0145 + 0.1 C_{L_{\alpha, W}}^2 \alpha_0^2)$$

## 1.5 Finding $L_H$

$$L_H = \eta_H \frac{S_H}{S_W} C_{L_H}$$

## Finding $C_{L_H}$

$$C_{L_H} \equiv C_{L_H}(\alpha, \epsilon, i_H)$$

1.  $\epsilon$  is also a function of  $\alpha$

2.  $C_{L_H} \approx C_{L_{H0}} + C_{L_{\alpha, H}} \alpha_H + C_{L_{i_H}} i_H |C_{L_{H0}} = 0, \text{ symmetry}$

3. set  $\alpha_H = \alpha_0 + i_H - \frac{d\epsilon}{d\alpha_W} \alpha_0$

4. - Find  $C_{L_{\alpha, H}}$  with the same process as the wing

(a) Find Taper ratio  $\lambda = \frac{2S_w}{bc_r}$

(b) Find SweepAngle of the LE:  $\Lambda_{LE} = \tan^{-1} \left( \tan \Lambda_{c/4} + \frac{1}{AR} \frac{1-\lambda}{1+\lambda} \right)$

(c) Find SweepAngle of the c/2:  $\Lambda_{c/2} = \tan^{-1} \left( \tan \Lambda_{LE} - \frac{2}{AR} \frac{1-\lambda}{1+\lambda} \right)$

(d) Find  $C_{l_\alpha} = \frac{2\pi}{\sqrt{1-M_\infty^2}}$

(e) Find Beta/k constants:  $\beta = \sqrt{1-M^2}$ ,  $k = \frac{C_{l_\alpha}}{2\pi}$

(f) plug into equation:

$$C_{L_{\alpha, H}} = \frac{2\pi AR}{2 + \sqrt{\frac{AR^2 \beta^2}{k^2} \left( 1 + \frac{\tan^2 \Lambda_{c/2}}{\beta^2} \right)} + 4}$$

5. Find  $C_{L_{i_H}}$  from the given equation:  $C_{L_{i_H}} = C_{L_{\alpha, H}} \eta_H \frac{S_H}{S_W}$

$$\therefore L_H = \eta_H \frac{S_H}{S_W} C_{L_{\alpha, H}} \left( \alpha_0 + i_H - \frac{d\epsilon}{d\alpha_W} \alpha_0 \right) + C_{L_{\alpha, H}} \eta_H \frac{S_H}{S_W} i_H$$

## 1.6 Finding $F_{N_0}$

Using the given equation

$$F_{N_0} = 2q_\infty A_{inlet} \cos \alpha_0^2 \sin \alpha_0$$

$A_{inlet}$  is given as  $15 \text{ ft}^2$  or  $1.39355 \text{ m}^2$

## 1.7 Finding $\epsilon$

$\epsilon$  estimated purely as a function of  $\alpha$

$$\epsilon = \frac{d\epsilon}{d\alpha_W}(\alpha_W - \alpha_0)$$

where  $\alpha_0$  is 0 because it is a symmetrical airfoil

$$\therefore \epsilon = \frac{d\epsilon}{d\alpha}\alpha_0$$

Then with only 3 unknowns remaining of  $T_0, \alpha_0, i_H$ , we used the SciPy fsolve function to solve the system of nonlinear equations.

## 2 Task 1b

Determine the most forward c.g. location from the tail incidence angle limit condition.

### 2.1 Equations to consider

$$CL(a, ih) = CL_{\alpha_{wf}} \cdot a + n_H \cdot CL_{\alpha_H} \cdot (a \cdot (1 - \delta_e \delta_a) + ih) \cdot \frac{S_H}{S_W} \quad (1)$$

$$CL_H(a, ih) = CL_{\alpha_H} \cdot (a \cdot (1 - \delta_e \delta_a) + ih) \quad (2)$$

$$CL_{wf}(a) = CL_{\alpha_{wf}} \cdot a \quad (3)$$

$$CM(a, ih) = CL_{wf}(a) \cdot (X_{cg} - X_{ac_{wf}}) - n_H \cdot CL_H(a, ih) \cdot (X_{ac_H} - X_{cg}) \cdot \frac{S_H}{S_W} \quad (4)$$

$$CL_{\min} = \frac{W}{q_{\inf} \cdot S_W} \quad (5)$$

## 2.2 Algorithm

The method to find the forward most position of the center of gravity uses the following algorithm:

### 1. Initialization:

- Set the minimum coefficient of lift  $CL_{\min}$  based on aircraft weight ( $W$ ), dynamic pressure ( $q_{\inf}$ ), and wing reference area ( $S_W$ ).
- Generate an array  $a$  with 1000 elements, ranging from -0.2 to 0.2.

### 2. Iterative Process:

- Iterate over a range of values for  $i$  from 0 to 26.
- Initialize empty lists for  $CL_{\text{list}}$  and  $CM_{\text{list}}$ .
- Calculate the midpoint ( $x_{cg}$ ) between  $x_{cg_{\text{high}}}$  and  $x_{cg_{\text{low}}}$ .
- Set lower and upper bounds ( $ilow$  and  $iup$ ) for a pitch angle ( $ih$ ) within a specified range.
- Calculate a guess for  $ih$  ( $guess_i h$ ) as the midpoint between  $ilow$  and  $iup$ .
- Iterate over the range of  $a$  values:
  - Calculate  $CL$  and  $CM$  for each  $a$  and  $ih$ .
  - Append these values to  $CL_{\text{list}}$  and  $CM_{\text{list}}$ .
- Find the index ( $l$ ) corresponding to the minimum value in  $CM_{\text{list}}$ .
- Calculate the angle of attack ( $alpha$ ) and the guessed  $CL$  value ( $CL_{\text{guess}}$ ) based on the minimum  $CM$ .
- Adjust the bounds for  $ih$  based on  $CL_{\text{guess}}$  relative to  $CL_{\min}$ .
- Adjust  $x_{cg_{\text{high}}}$  or  $x_{cg_{\text{low}}}$  based on the magnitude of  $ih$  if it's less than or greater than 20 degrees.

### 3 Task 1 Tables

#### Givens

	Wing	Horizontal Tail (all moveable)	Vertical Tail	Fuselage
Area ( $S$ )	400 ft <sup>2</sup>	100 ft <sup>2</sup>	70 ft <sup>2</sup>	Length = 48 ft
Root chord ( $c_r$ )	17 ft	10 ft	10 ft	
Aspect Ratio ( $AR$ )	$AR_W = 3.6$	$AR_H = 3.4$	$AR_V = 1.5$	
Span ( $b$ )	$b = \sqrt{(AR)S_W}$ = 37.947	$b = \sqrt{(AR)S_H}$ = 18.439	$b = \sqrt{(AR)S_V}$ = 10.247	
Quarter chord sweep ( $\Lambda_c$ ) +	45°	45°	45°	Max fuselage width = 6 ft
Airfoil Lift curve slope	$c_{l_{\alpha_W}} = 5.6$	$c_{l_{\alpha_H}} = 6$	$c_{l_{\alpha_V}} = 6$	
Incidence angle	$i_W = 0$	$i_H = \text{variable} \pm 20^\circ$	--	
Tail efficiency factor	--	$\eta_H = 0.5$	$\eta_V = 1.0$	Volume = 1000 ft <sup>3</sup>
Fuselage interference factor ( $k$ )	--	--	$k = 1.0$	
X-apex	16 ft	36 ft	35 ft	
Z-apex	0	--	--	
Downwash/Sidewash gradient	--	$\frac{d\varepsilon}{d\alpha} = 0.75$	$\frac{d\sigma}{d\beta} = 0.12$	$X_{AC_{WF}} = 0.05\bar{c}_w$ (Its 0.05 in front of the wing)
$X_{AC}$	$\frac{0.32\bar{c}_w}{\bar{c}_w} = 11.86$	$\frac{0.32\bar{c}_w}{\bar{c}_H} = 6.71$	$\frac{0.32\bar{c}_w}{\bar{c}_V} = 7.32$	

Table 1. Trim Analysis, Nominal Stability, and Control Derivatives

Parameter/Non-dimensional Stability Derivative	Equations used or Reference (if applicable)	Value (units)
$\alpha_0$	See Code	10.6284°
$i_H$	See Code	-4.9097°
$C_{L_0}(Trim)$	$C_{L_W} + \eta \frac{S_H}{S_W} C_{L_H} \cos \varepsilon$	0.6572
$C_{D_0}(Trim)$	$C_D = 0.0145 + 0.1 C_{L_W}^2$ $C_{D_0} = 0.0145 + 0.1 \left( C_{L_{\alpha_W}} (\alpha_{trim} - \alpha_{zero-lift}) \right)^2$ $C_{D_0} = 0.0145 + 0.1 \left( C_{L_{\alpha_W}} \alpha_{trim} \right)^2$	0.04836
Trim thrust	See Code	1849.2467
Normal Force	$F_N = \dot{m}_p V_\infty \cos \alpha \sin \alpha$ $= \rho A_{in} (V_\infty \cos \alpha)^2 \sin \alpha$ $= 2q_\infty A_{in} \cos^2 \alpha \sin \alpha$	499.7178
Propulsive $C_{P_{M_0}}$	$F_{N_0} (X_{CG} - 0) / (q_\infty \bar{c}_W S_W)$	0.03042
$C_{M_0}(Trim, Aerodynamic)$	$-C_{P_{M_0}}$	-0.03042
$C_{L_\alpha}$	$C_{L_{\alpha_W \& F}} + C_{L_{\alpha_H}} \eta_H \frac{S_H}{S_W} \left( 1 - \frac{d\varepsilon}{d\alpha} \right)$	3.2397
$C_{D_\alpha}$	$C_D = 0.0145 + 0.1 C_{L_W}^2$ $C_D = 0.0145 + 0.1 \left( C_{L_{\alpha_W}} (\alpha - \alpha_{zero-lift}) \right)^2$ $C_{D_\alpha} = 0.2 C_{L_{\alpha_W}}^2 \alpha$	0.3652
$C_{L_{i_H}}$	$C_{L_{i_H}} = C_{L_{\alpha_H}} \eta_H \left( \frac{S_H}{S_W} \right)$	0.4096
$\bar{X}_{NP}$	$\bar{X}_{NP} = \frac{C_{L_{\alpha_W F}} \bar{X}_{ACWF} + C_{L_{\alpha_H}} n_H \left( \frac{S_H}{S_W} \right) \bar{X}_{ACH} \left( 1 - \frac{d\varepsilon}{d\alpha} \right)}{C_{L_\alpha}}$	2.4018
$SM$	$SM = -\frac{C_{M_\alpha}}{C_{L_\alpha}}$	0.1252
$C_{M_\alpha}$	$C_{M_\alpha} = C_{L_{\alpha_W F}} \Delta \bar{X}_{ACWF} - C_{L_{\alpha_H}} \eta_H V_H \left( 1 - \frac{d\varepsilon}{d\alpha} \right)$	-0.4058
$C_{M_{i_H}}$	$C_{M_{i_H}} = -C_{L_{\alpha_H}} \eta_H V_H$	-0.5278
Forward CG limit	Current CG at 27 ft	13.5



$ V_V $	$M = \frac{U_\infty}{a} \rightarrow$ $U_\infty = 0.52 * 660 \text{ mph} = 343.2 \text{ mph}$	$503.36 \frac{ft}{s}$
$q_\infty$	$\frac{1}{2} \rho V_V^2 = \frac{1}{2} \left( 0.000738 \frac{slugs}{ft^3} \right) \left( 503.36 \frac{ft}{s} \right)^2$	93.494
$\lambda$	$\frac{2b - c_r AR}{c_r AR}$	$\lambda_w = 0.24085$ $\lambda_H = 0.084647$ $\lambda_V = 0.366267$
$\bar{c}$	$\frac{2(1 + \lambda + \lambda^2)}{3(1 + \lambda)} c_r$	$\bar{c}_w = 11.86$ $\bar{c}_H = 6.71$ $\bar{c}_V = 7.32$
$x_{LEMAC}$	$\frac{b}{2} \left( \frac{1 + 2\lambda}{3(1 + \lambda)} \right) \tan(\Lambda_{LE})$	Wing = 8.83299 HT = 4.1353 VT = 2.8362
$X_{ACW}, \overline{X_{ACW}}$	$\frac{X_{APPW} + x_{LEMAC} + 0.32\bar{c}_w,}{\bar{c}_w}$ $\frac{X_{APPW} + x_{LEMAC} + 0.32\bar{c}_w}{\bar{c}_w}$	28.628, 2.41383
$X_{ACWF}, \overline{X_{ACWF}}$	$\frac{X_{APPW} + x_{LEMAC} + 0.27\bar{c}_w,}{\bar{c}_w}$ $\frac{X_{APPW} + x_{LEMAC} + 0.27\bar{c}_w}{\bar{c}_w}$	28.0352, 2.36284
$X_{ACH}, \overline{X_{ACH}}$	$\frac{X_{APPH} + x_{LEMAC} + 0.32\bar{c}_H,}{\bar{c}_w}$ $\frac{X_{APPH} + x_{LEMAC} + 0.32\bar{c}_H}{\bar{c}_w}$	42.2825, 3.56513
$C_{L_{\alpha\#}}$	$\frac{2\pi AR}{2 + \sqrt{\frac{AR^2 \beta^2}{k^2} \left( 1 + \frac{\tan^2(\Lambda_c)}{\beta^2} \right) + 4}}$	$C_{L_{\alpha WF}} = C_{L_{\alpha W}} = 3.1373$ $C_{L_{\alpha H}} = 3.277$
$\Delta \bar{X}_{ACH}$	$\frac{X_{ACH} - X_{Ref(CG)}}{\bar{c}_w}$	1.288
$\Delta \bar{X}_{ACWF}$	$\frac{X_{Ref(CG)} - X_{ACW\&F}}{\bar{c}_w}$	-0.087285
$k$	$\frac{c_{l\alpha}}{2\pi}$	1.17073
$\Lambda_{LE}$	$\arctan \left( \frac{AR \tan \left( \Lambda_{\frac{1}{4}} \right) + \frac{1 - \lambda}{1 + \lambda}}{AR} \right)$	$\Lambda_{LEW} = 49.47^\circ$ $\Lambda_{LEH} = 51.30^\circ$ $\Lambda_{LEV} = 52.63^\circ$
$\Lambda_{1/2}$	$\arctan \left( \frac{AR \tan \left( \Lambda_{\frac{1}{4}} \right) - 4 \left( \frac{1}{2} - \frac{1}{4} \right) \left( \frac{1 - \lambda}{1 + \lambda} \right)}{AR} \right)$	$\Lambda_{w(1/2)} = 39.69^\circ$ $\Lambda_{h(1/2)} = 36.94^\circ$ $\Lambda_{v(1/2)} = 34.64^\circ$

## 4 Task 2

### Summary of Longitudinal Stability and Control Derivatives

#### 1. $C_{D_M}$ and $C_{L_M}$ : Drag and Lift Contributions due to Pitching Motion

$$C_{D_M} = \frac{M \cos^2(q_{c_W})}{1 - M^2 \cos^2(q_{c_W})} \cdot C_{D0}$$

$$C_{L_M} = \frac{M \cos^2(q_{c_W})}{1 - M^2 \cos^2(q_{c_W})} \cdot C_{L0}$$

#### 2. $C_{M_q}$ : Pitch Damping Derivative

$$C_{M_q} = -n_H \cdot C_{L_{a_H}} \cdot \frac{S_H}{S_W} \cdot (X_{ac_h} - X_{cg})^2 \cdot \cos^2(\alpha) \cdot \frac{1}{U_0 \cdot c_{w_w}}$$

#### 3. $C_{M_{\dot{a}}}$ : Rate of Change of Pitch Damping with Respect to Angle of Attack

$$C_{M_{\dot{a}}} = -n_H \cdot \frac{S_H}{S_W} \cdot C_{L_{a_H}} \cdot (X_{ac_h} - X_{cg}) \cdot (X_{ac_h} - X_{ac_W}) \cdot \cos^2(\alpha) \cdot \delta_e / \delta_a$$

#### 4. $X_u, Z_u, M_u$ : Longitudinal Stability Derivatives

$$X_u = -\frac{(C_{D_u} + \frac{2}{U_0} \cdot C_{D0}) \cdot q_\infty \cdot S_W}{m}$$

$$Z_u = -\frac{(C_{L_u} + \frac{2}{U_0} \cdot C_{L0}) \cdot q_\infty \cdot S_W}{m}$$

$$M_u = -\frac{(C_{M_u} + \frac{2}{U_0} \cdot \text{-prop\_CP\_M0}) \cdot q_\infty \cdot S_W \cdot c_{w_w}}{I_{yy}}$$

#### 5. $X_a, Z_a, M_a$ : Longitudinal Control Derivatives

$$X_a = -\frac{(-C_{D_\alpha} + C_{L0}) \cdot q_\infty \cdot S_W}{m}$$

$$Z_a = -\frac{(-C_{L_\alpha} + C_{D0}) \cdot q_\infty \cdot S_W}{m}$$

$$M_a = \frac{C_{M_\alpha} \cdot q_\infty \cdot S_W \cdot c_{w_w}}{I_{yy}}$$

#### 6. $M_q$ : Pitch Control Derivative

$$M_q = \frac{C_{M_q} \cdot q_\infty \cdot S_W \cdot c_{w_w}}{I_{yy}}$$

Table 2. Other Longitudinal Nondimensional and Dimensional Stability Derivatives

Stability Derivative	Equations Used	Values (units)
$C_{D_M}$	$\frac{M_\infty \cos^2 \Lambda_{W_{c/4}}}{1 - M_\infty^2 \cos^2 \Lambda_{W_{c/4}}} C_{D_0}$	0.01454
$C_{L_M}$	$\frac{M_\infty \cos^2 \Lambda_{W_{c/4}}}{1 - M_\infty^2 \cos^2 \Lambda_{W_{c/4}}} C_{L_0}$	0.1975
$C_{M_u}$	Neglect	0
$C_{M_q}$	$-\eta_H \frac{S_H}{S_W} C_{L_{\alpha_H}} (X_{AC_H} - X_{CG})^2 \cos^2 \alpha_0 \frac{1}{U_0 \bar{c}_W}$	-0.01548
$C_{M_\alpha}$	$-\eta_H \frac{S_H}{S_W} C_{L_{\alpha_H}} (X_{AC_H} - X_{CG})(X_{AC_H} - X_{AC_W}) \cos^2 \alpha_0 \frac{d\varepsilon}{d\alpha} \frac{1}{U_0 \bar{c}_W}$	-0.01037
$X_u$	$\frac{-(C_{D_u} + \frac{2}{U_0} C_{D_0}) q_\infty S_W}{m}$	-0.01134
$Z_u$	$\frac{-(C_{L_u} + \frac{2}{U_0} C_{L_0}) q_\infty S_W}{m}$	-0.1541
$M_u$	$\frac{-(C_{M_u} + \frac{2}{U_0} C_{M_0}) q_\infty S_W \bar{c}_W}{I_{yy}}$	0.001729
$X_\alpha$	$\frac{(-C_{D_\alpha} + C_{L_0}) q_\infty S_W}{m}$	15.9862
$Z_\alpha$	$\frac{-(C_{L_\alpha} + C_{D_0}) q_\infty S_W}{m}$	-179.977
$M_\alpha$	$\frac{C_{M_\alpha} q_\infty S_W \bar{c}_W}{I_{yy}}$	-5.8059
$M_q$	$\frac{C_{M_q} q_\infty S_W \bar{c}_W}{I_{yy}}$	-0.2215
$M_{\dot{\alpha}}$	$\frac{C_{M_{\dot{\alpha}}} q_\infty S_W \bar{c}_W}{I_{yy}}$	-0.1484

$C_{D_u}$	$\frac{C_{D_M}}{a_0}$	1.5022 e -05
$C_{L_u}$	$\frac{C_{L_M}}{a_0}$	0.0002041

## 4.1 Analysis of the Longitudinal Derivatives

$$\mathbf{A}_{\text{mat}} = \begin{bmatrix} \frac{X_u}{U_0} & \frac{X_a}{U_0} & 0 & -32.2 \\ \frac{Z_u}{U_0} & \frac{Z_a}{U_0} & 1 & 0 \\ \frac{M_u + M_a \dot{Z}_u}{U_0} & \frac{M_a + M_a \dot{Z}_a}{U_0} & M_q + M_a \dot{Z}_a & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$A = \text{eig.real}$$

$$B = \text{eig.imag}$$

$$\text{natural\_freq} = \sqrt{A^2 + B^2}$$

$$\text{damping\_ratio} = -\frac{A}{\text{natural\_freq}}$$

$$\text{time\_to\_half} = \frac{\log 2}{|A|}$$

$$\text{time\_constant} = \frac{1}{|A|}$$

$$\text{cycle\_to\_half} = \left| \frac{\text{time\_to\_half}}{2\pi/B} \right|$$

## 4.2 Explanation of Dynamic Parameters from Eigenvalues

The eigenvalues of the longitudinal derivatives matrix provide valuable information about the dynamic behavior of the system. From these eigenvalues ( $A$  and  $B$ ), several key parameters are derived:

### Natural Frequency ( $\omega_n$ )

The natural frequency is determined from the real and imaginary parts of the eigenvalues using the formula:

$$\text{natural\_freq} = \sqrt{A^2 + B^2}$$

It represents the rate at which the system oscillates without damping when disturbed.

### Damping Ratio ( $\zeta$ )

The damping ratio is calculated as the negative ratio of the real part to the natural frequency:

$$\text{damping\_ratio} = -\frac{A}{\text{natural\_freq}}$$

It provides information about the rate at which the system's amplitude decreases over time.

### Time to Half Amplitude ( $t_{1/2}$ )

The time to half amplitude is computed as the natural logarithm of 2 divided by the absolute value of the real part of the eigenvalue:

$$\text{time\_to\_half} = \frac{\log 2}{|A|}$$

It indicates the time it takes for the system's amplitude to decrease to half of its initial value.

### Time Constant ( $\tau$ )

The time constant is the reciprocal of the absolute value of the real part of the eigenvalue:

$$\text{time\_constant} = \frac{1}{|A|}$$

It represents the time required for the system's response to reach  $1 - \frac{1}{e}$  (approximately 63.2

## Cycles-to-Half ( $n_{1/2}$ )

Cycles-to-half is calculated by taking the absolute value of the time to half amplitude and dividing it by the period of oscillation:

$$\text{cycle\_to\_half} = \left| \frac{\text{time\_to\_half}}{2\pi/B} \right|$$

It gives the number of oscillations required for the system's amplitude to decrease to half.

These parameters provide insights into the dynamic behavior of the system, crucial for stability and control analysis in aerospace engineering.

## 4.3 Categorizing the Aircraft

By leveraging the content presented in the lecture slides, we gain the capability to systematically classify aircraft according to their performance and stability characteristics. The comprehensive insights provided in these educational materials allow us to analyze and categorize various aircraft models, taking into consideration factors such as aerodynamic performance, longitudinal and lateral stability, and other crucial parameters. This categorization is instrumental in enhancing our understanding of different aircraft types, their design principles, and their operational capabilities, thereby contributing to a more profound comprehension of aeronautical engineering principles and the broader field of aviation.

## 4.4 Task 2 Tables

*Table 3. Modal Characteristics (4x4 system, no approximations needed)*

Mode	Natural Frequency (rad/s)	Damping Ratio	Damped Frequency (rad/s)	Time-to-half (s)	Cycles-to-half	Time Constant (s)
Phugoid	0.1145	0.06523	0.1145	92.7219	1.6875	133.7695
Short Period	2.4237	0.1493	2.3965	1.9151	0.7304	2.7630

*Table 4. Flying Qualities (Airplane Class, Flight Phase)*

Parameter	Value (units)	FQ Level
Phugoid damping ratio	0.06523	Level 1 Cat. B
Short Period damping ratio	0.1493	Level 3 Cat. B
Short Period Natural Frequency	2.4237 (rad/s)	Level 1 Cat. B

```
##### UNCHANGED TASK 1 CODE #####
from scipy.optimize import fsolve
import numpy as np
from sympy import (symbols, sin, cos, solve, pi)
import math
a_0 = symbols('a_0')
T_0 = symbols('T_0')
i_h = symbols('i_h')
#Variables

W = 22000 #lb
A_in = 15 # sqft

cw_w=11.86
cw_h=6.71
cw_v=7.32

X_ac_wf = 28.0352 #ft
X_ac_h = 42.2825 #ft
X_ac_w = 28.628 #ft
X_cg = 27 #ft

S_H = 100 #ft^2
S_W = 400 #ft^2
A_in = 15 #ft^2

n_H = 0.5
rho=7.38*(10**-4) # slug/ft^3
q_inf = (1/2) * rho * (503.36 ** 2) #whatever units pressure is
#print(q_inf)
q_H = n_H * q_inf
vv=503.36 #ft/s

CL_a_wf = 3.1373
CL_a_H = 3.277
CL_a_W = 3.1373 #5.7875

de_da = 0.75

# Function representing the system of equations

def equations(variables):
    a_0, i_h, T_0 = variables

    e = de_da * a_0
    a_H = a_0 + i_h - e
    L_wf = CL_a_W * a_0 * q_inf * S_W #CL_a_wf is the same as CL_a_w
    L_h = CL_a_H * a_H * q_H * S_H
    C_D0 = 0.0145 + 0.1*((CL_a_W*a_0)**2)
    D_0 = q_inf * S_W * C_D0
    F_N0 = 2 * q_inf * A_in * (np.cos(a_0)**2) * np.sin(a_0)

    # Trim equations
    #X = T_0 - W*np.sin(a_0) + L_wf * np.sin(a_0) - D_0 * np.cos(a_0) + L_h*np.sin(a_0 - e)
    X = T_0 * np.cos(a_0) - F_N0 * np.sin(a_0) - D_0 - L_h * np.sin(e)

    #Z = W*np.cos(a_0) - F_N0 - L_wf * np.cos(a_0) - D_0 * np.sin(a_0) - L_h* np.cos(a_0 - e)
    Z = W - T_0 * np.sin(a_0) - F_N0 * np.cos(a_0) - L_wf - L_h * np.cos(e)

    #M = F_N0 * (X_cg - 0) - L_wf * (X_ac_wf - X_cg)*np.cos(a_0) - D_0 * (X_ac_w - X_cg) * np.sin(a_0) - L_h * (X_ac_h - X_cg) * np.cos(a_0 - e)
    M = F_N0 * (X_cg - 0) - L_wf * (X_ac_wf - X_cg)*np.cos(a_0) - D_0 * (X_ac_w - X_cg) * np.sin(a_0) - L_h * (X_ac_h - X_cg) * np.cos(a_0 - e)

    return [X, Z, M]

# Initial guess for the variables
initial_guess = [np.radians(10), np.radians(-5), 2000] # Provide your initial guess here
radtodeg = 180/np.pi
# Solve the equations
solution = fsolve(equations, initial_guess)

print("Alpha trim is ", solution[0]* radtodeg, '\n')
print("Incidence Angle is ", (solution[1]) * radtodeg, "\n")
print("Trim Thrust is ", solution[2], "\n")

alpha=solution[0]
i_h=solution[1]
thrust=solution[2]
```

```

Alpha trim is 10.628368614640332

Incidence Angle is -4.909762693703019

Trim Thrust is 1849.2467118310285

#Most Forward CG
def CL(a,ih):
    return CL_a_wf * a + n_H * CL_a_H * (a * (1-de_da) + ih)*(S_H/S_W)

def CL_H(a, ih):
    return CL_a_H * (a*(1 - de_da) + ih)

def CL_wf(a):
    return CL_a_wf * a

def CM(a,ih):
    return CL_wf(a) * (X_cg - X_ac_wf) - n_H * CL_H(a,ih)* (X_ac_h - X_cg) * (S_H/S_W)

CL_min = W / (q_inf * S_W)

a = np.linspace(-0.2,0.2,1000)
xcg_high = 27
xcg_low = 0
for i in range(27):
    CL_list = []
    CM_list = []
    xcg = (xcg_high+xcg_low)/2
    ilow = -25*np.pi/180
    iup = 15*np.pi/180
    guess_ih = (iup+ilow)/2
    for j in range(len(a)):
        CL_list.append(CL(a[j],guess_ih))
        CM_list.append(abs(CM(a[j],guess_ih)))
    l = CM_list.index(min(CM_list))
    alpha = a[l]*180/np.pi
    CLguess = CL_list[l]
    if CLguess < CL_min:
        iup = guess_ih
        ilow = ilow
    if CLguess > CL_min:
        ilow = guess_ih
        iup = iup
    if abs(guess_ih*180/np.pi) < 20:
        xcghigh = xcg
    if abs(guess_ih*180/np.pi) > 20:
        xcglow = xcg

print('Forward Most CG Location',xcg)
print()

```

 Forward Most CG Location 13.5

Double-click (or enter) to edit

```

#Rest of Table 1
alpha = solution[0]
e = de_da * alpha
AR_w=3.6
M=0.52
beta=math.sqrt(1-M**2)

c_la=7.35593
k=c_la/(2*math.pi)

half_chord=39.69

Normal_force = 2 * q_inf * A_in * np.cos(alpha) ** 2 * np.sin(alpha)
print("The normal force is", Normal_force, "\n")

```

```

C_D0 = 0.0145 + 0.1*((CL_a_W * alpha)**2)
print("C_D0", C_D0, "\n")

D_0 = q_inf * S_W * C_D0

CL_L0 = CL_a_wf*alpha + n_H*(S_H/S_W)*CL_a_H*alpha*np.cos(e)
print("C_L0", CL_L0, "\n")

prop_CP_M0=(Normal_force*(X_cg-0))/(q_inf*cw_w*S_W)
print("Propulsive force", prop_CP_M0, "\n")

#work=math.sqrt(((3.6**2 * beta**2) /k**2) * (1+((math.tan(math.radians(half_chord))**2)/beta**2))+4)
#CL_alpha=(2*math.pi*3.6)/(2 + work)
CL_alpha = CL_a_wf + CL_a_H * n_H * (S_H / S_W) * (1 - de_da)
print("Big CL_a", CL_alpha, "\n")

deltaX_ach = (X_ac_h - X_cg)/ cw_w
deltaX_acwf = (X_cg - X_ac_wf)/ cw_w

CM_a = CL_a_wf*deltaX_acwf - CL_a_H * n_H * (S_H/S_W) * deltaX_ach * (1- de_da)
print("C_Ma = ", CM_a, "\n")

CM_ih = -CL_a_H*n_H * (S_H/S_W) * deltaX_ach
print("CM_ih = ", CM_ih, "\n")

C_Da=0.2*CL_a_W**2*alpha
print("C_Da", C_Da, "\n")

X_bar_np=(CL_a_wf * (X_ac_wf/cw_w ) + CL_a_H*n_H*(S_H/S_W)*(X_ac_h/cw_w)*(1-de_da))/CL_alpha
print("X_bar_np", X_bar_np, "\n")

SM=-CM_a/CL_alpha
print("SM", SM, "\n")

CM_ih=-CL_a_H*n_H*((S_H/S_W) * deltaX_ach)
print("CM_ih", CM_ih, "\n")

CL_ih=CL_a_H*n_H*(S_H/S_W)
print("CL_ih", CL_ih, "\n")

The normal force is 499.7177220566599

C_D0 0.0483688188104683

C_L0 0.6572204881969858

Propulsive force 0.03042006007273037

Big CL_a 3.2397062500000002

C_Ma = -0.40579734195826306

CM_ih = -0.5278325516441822

C_Da 0.3651624149091098

X_bar_np 2.4018173145785666

SM 0.12525744948581774

CM_ih -0.5278325516441822

CL_ih 0.409625

```



```

# Task 2
import math
import numpy

q_c_W=np.radians(45)
q_c_H=np.radians(45)
q_c_V=np.radians(45)
M=0.52
U_0=503.36 # ft/s
C_Mu=0
I_xx=7210 # slug-ft^2
I_yy=31000 # slug-ft^2
I_zz=37000 # slug-ft^2
I_xz=1000 # slug-ft^2
m=22000/32.2
sos = 968 #speed of sound at cruising altitude (ft/s)

C_DM= ((M*math.cos(q_c_W)**2)/(1 - (M**2) * math.cos(q_c_W)**2)) * C_D0
print("C_DM", C_DM, "\n")

C_LM=((M*math.cos(q_c_W)**2)/(1 - (M**2) *math.cos(q_c_W)**2)) * C_L0
print("C_LM", C_LM, "\n")

C_Mu=0
print("C_Mu", C_Mu, "\n")

C_Mq= -n_H * CL_a_H* (S_H/S_W) * (X_ac_h-X_cg)**2 * (math.cos(alpha)**2) *(1/(U_0*cw_w))
print("C_Mq", C_Mq, "\n")

C_Ma_dot= -n_H * (S_H/S_W) * CL_a_H * (X_ac_h-X_cg) * (X_ac_h-X_ac_w) * (math.cos(alpha)**2) * de_da*(1/(U_0*cw_w))
print("C_Ma_dot", C_Ma_dot, "\n")

C_Du = C_DM / sos
X_u=-((C_Du+(2/U_0)*C_D0)*q_inf*S_W)/m
#X_u = (-q_inf * S_W * (2*C_D0)) / (m*U_0) #Maybe this equation
print("X_u", X_u, "\n")

C_Lu = C_LM / sos

Z_u=-((C_Lu + (2/U_0) * C_L0)* q_inf * S_W)/m
#Z_u = (-q_inf * S_W * (2*C_L0)) / (m*U_0)
print("Z_u", Z_u, "\n")

M_u=-((C_Mu+(2/U_0)*-prop_CP_M0)*q_inf*S_W*cw_w)/I_yy
print("M_u", M_u, "\n")

X_a=(-C_Da+C_L0)*q_inf*S_W)/m
print("X_a", X_a, "\n")

Z_a=-((CL_alpha+C_D0)*q_inf*S_W)/m
print("Z_a", Z_a, "\n")

M_a=(C_Ma*q_inf*S_W*cw_w)/I_yy
print("M_a", M_a, "\n")

M_q=(C_Mq*q_inf*S_W*cw_w)/I_yy
print("M_q", M_q, "\n")

M_a_dot=(C_Ma_dot*q_inf*S_W*cw_w)/I_yy
print("M_a_dot", M_a_dot, "\n")

C_DM 0.014541966802407215

C_LM 0.19759172864386718

C_Mu 0

C_Mq -0.015480364668898882

C_Ma_dot -0.010373465043586445

X_u -0.01134175628262406

Z_u -0.15410826198352734

M_u 0.001729330959888878

```

```
X_a = 15.98623401381166

# Task 3
import numpy as np
import math
from numpy.linalg import eig

A_mat=np.array([[X_u, X_a, 0, -32.2],
                [(Z_u / U_0), (Z_a / U_0), 1, 0],
                [M_u + (M_a_dot * Z_u) / U_0, M_a + (M_a_dot * Z_a) / U_0, (M_q + M_a_dot), 0],
                [0, 0, 1, 0]])
```