Minimum-Time Aircraft Trajectory Optimization using Full-Space and Reduced-Space Methods

Erin E. McNeil[1]
Georgia Institute of Technology 1, Atlanta, GA, 30332, U.S.A.

This project investigates a minimum-time-to-climb trajectory optimization problem for a supersonic aircraft using both full-space and reduced-space numerical formulations. The trajectory is governed by nonlinear dynamics incorporating surrogate models for aerodynamic coefficients, atmospheric properties, and engine thrust. To enable efficient optimization, six surrogate models are developed: thrust (linear plane-fit over a Delaunay triangulation), lift slope (C_{L_α}), drag coefficient (C_{D_0}), air density (ρ), speed of sound (a), and drag rise factor (η) (via piecewise and spline fits). Each surrogate model computes its local state and control derivatives via the complex-step method for machine-precision sensitivities. The reduced-space formulation treats the control history (angle of attack α) and final time t_f as design variables, with the state trajectory integrated using the implicit trapezoidal method and solved via Newton iterations. An adjoint-based sensitivity method is implemented to compute gradients of the final-state constraint residuals with respect to the design vector. The full-space formulation, in contrast, treats all state and control values as design variables and enforces the system dynamics and boundary conditions as constraints. Its Jacobian is formed with complex-step differentiation and then validated against finite-difference directional-derivative checks. Results demonstrate that both formulations achieve high-precision constraint satisfaction and comparable final times (under 360 seconds). The reduced-space method achieves gradient errors below 1e-10 relative to complex-step benchmarks, while the full-space Jacobian maintains consistent derivative accuracy below 1e-9 compared to finite-difference. Key challenges included numerical instability in early reduced-space iterations and sensitivity of the optimizer to initial AOA guess structures. Clamping, scaling, and structured initial guesses were implemented to improve convergence. The reduced-space approach, using only a low-dimensional control basis, was very sensitive to the initial AOA-guess and often required more iterations (and time) to converge. In contrast, the full-space formulation, with per-node AOA variables, offered greater flexibility and tighter constraint enforcement, but at the cost of potentially unrealistic, highly oscillatory AOA profiles that may not reflect practical actuator limits.

I. Problem Formulation

The objective of this project is to minimize the time required for a supersonic aircraft to ascend from sea level to a specified final state (target velocity, flight path angle, and altitude). The aircraft's trajectory is governed by nonlinear dynamics with 5 state variables velocity, flight-path-angle, altitude, downrange, and mass (v, γ, h, r, m), and one control input (angle of attack, α).

The equations of motion are implemented in descriptor form:

E(q) · dq/dt = f(q(t), u(t))

where:

- q(t): state vector [v, gamma, h, r, m],
- u(t) = alpha(t): control input (angle of attack),
- f: right-hand-side nonlinear dynamics function,
- E(q) = I: identity matrix

The equations of motion are integrated using the trapezoidal method:

x_{k+1} - x_k - (dt/2)*[f(x_k, u_k) + f(x_{k+1}, u_{k+1})] = 0

where:

x = [v, gamma, h, r, m]^T,   u = [alpha]

[1] MS Aerospace Engineering, AE 6310: Optimization Design of Engineering Systems.

The continuous-time dynamics are defined as:

```
x0 = v_dot         = (T*cos(alpha) - D)/m - g*sin(gamma)
x1 = gamma_dot     = (T*sin(alpha))/(m*v) + L/(m*v) - (g*cos(gamma))/v
x2 = h_dot         = v*sin(gamma)
x3 = r_dot         = v*cos(gamma)
x4 = m_dot         = -T/(g*Isp)
```

Aerodynamic forces:

```
D = 0.5*rho*v^2*S*CD    (Drag force) (lbf)
L = 0.5*rho*v^2*S*CL     (Lift force) (lbf)

CD = CD0 + eta*CL_alpha*alpha^2
CL = CL_alpha*alpha

M = v/a

g    = 32.2 (ft/s^2)
S    = 530 (ft^2)
Isp  = 1600 (s)

T    = Thrust provided as a function of Mach and altitude (lbf)
CL_alpha = Lift curve slope provided as a function of Mach
CD0  = zero-lift drag coefficient provided as a function of Mach
Eta = drag rise factor provided as a function of Mach
rho  = atmospheric density provided as a function of altitude (slug/ft^3)
 a = speed of sound provided as a function of altitude (ft/s)
```

In the full-space formulation, all state and control variables at each time point are treated as design variables, with dynamic and terminal constraints enforced directly. The reduced-space formulation instead treats only α and final time as design variables, solving for the state trajectory via integration. Terminal constraints are imposed via adjoint sensitivities.

II. Surrogate Model Development

Six surrogate models were created to emulate aerodynamic and atmospheric data: Thrust, $C_{L\alpha}$, C_{D0}, Density (ρ), Speed of Sound (a), and Drag Rise Factor (η). These models were constructed using spline and interpolation methods, with complex-step-compatible logic. Each surrogate model was validated using training MSE and R^2:

Table 1: MSE and R^2 of Surrogate Models

Model	R^2	MSE
Thrust	1.0000	~0
C_L_alpha	>0.99	<0.01
C_D0	>0.99	<0.001
rho (Density)	>0.999	~0
a (Speed Sound)	>0.999	~0
eta	>0.95	<0.01

Each surrogate model was validated on the training dataset by computing the Mean Squared Error (MSE) and the coefficient of determination (R^2). Since the surrogates use interpolation and cubic spline techniques, the training MSE was near zero and R^2 was close to 1, indicating excellent fidelity to the original data. Figures 1–6 illustrate the fit quality and physical trends for each surrogate. These follow closely to the plots from the project's source data. [1] [2]

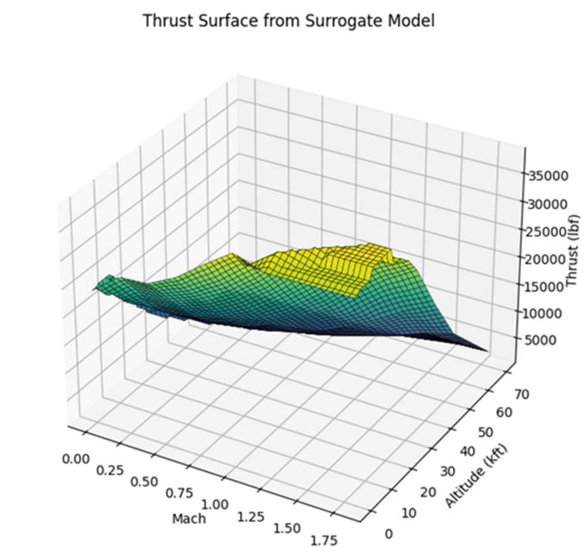

Figure 1: Surrogate model for Thrust

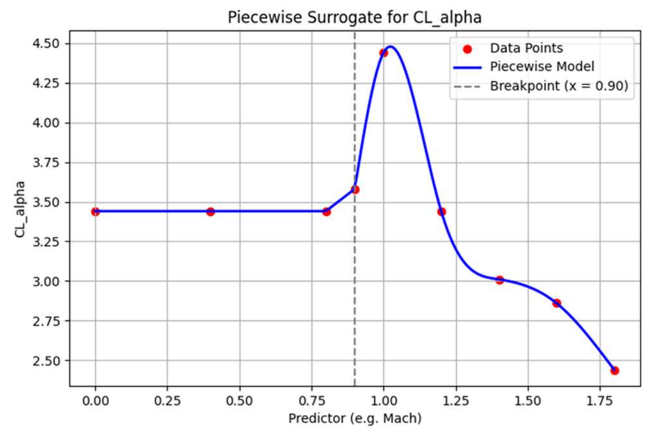

Figure 2: Surrogate model for C_Lalpha

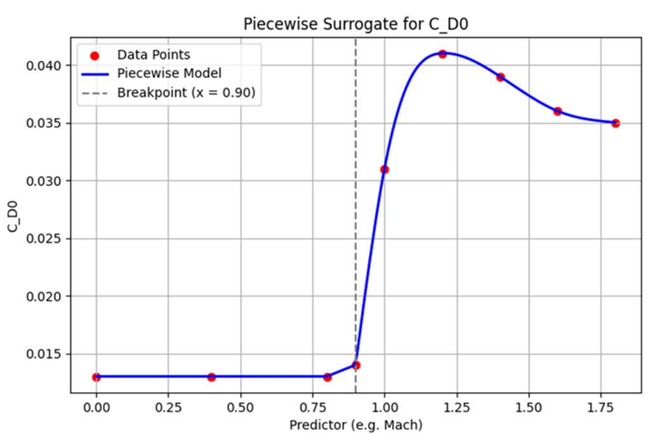

Figure 3: Surrogate model for C_D0

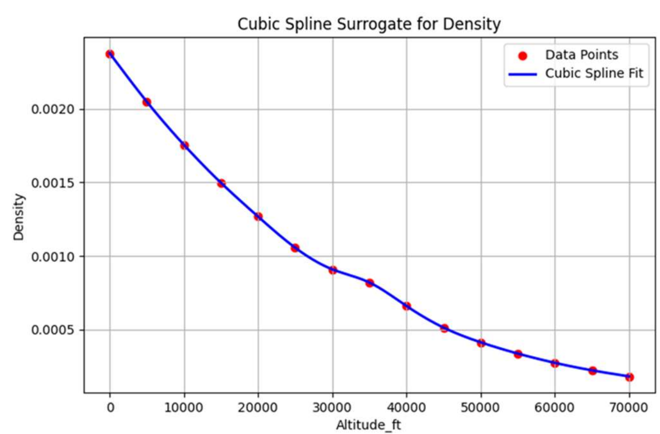

Figure 4: Surrogate model for Density

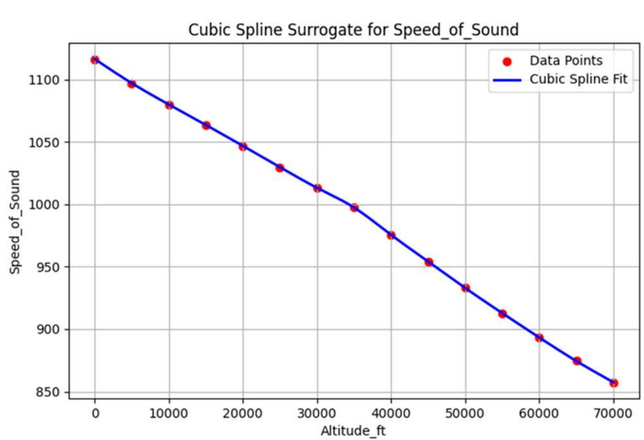

Figure 5: Surrogate model for Speed_of_Sound

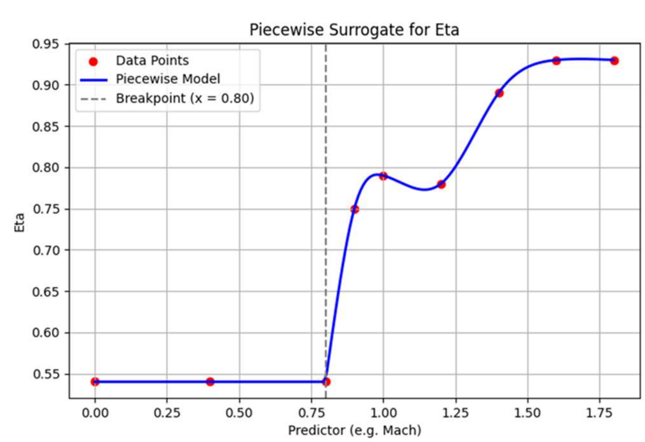

Figure 6: Surrogate model for Eta

III. Optimization Problem Formulation

This project focuses on solving a minimum-time trajectory optimization problem for an aircraft. The objective is to determine the optimal angle-of-attack trajectory and final time required to guide the aircraft from initial conditions to a prescribed final state, subject to realistic aerodynamic, atmospheric, and thrust models.

A. Reduced-Space Method and Jacobian Accuracy

The reduced-space trajectory optimizer integrates dynamics using the trapezoidal method and uses Newton's method at each time step. Final-state residuals are enforced as constraints, and their gradients are calculated using a reverse-time adjoint method. Complex-step verification of adjoint gradients showed relative errors less than 1e-10.

In the reduced-space approach, the states are implicitly computed from controls and dynamics using the implicit trapezoidal integration method:

$$q_{k+1} = q_k + \Delta t/2 \cdot [f(q_k, \alpha_k) + f(q_{k+1}, \alpha_{k+1})]$$

The design variables are the control points $\alpha(t)$ and the final time tf. The constraints are imposed only on the final state:

$$c(x) = (q_{final} - q_{target}) / scale = 0$$

This method results in fewer variables and enables the use of adjoint-based sensitivity analysis for efficient gradient computation.. I derived the adjoint derivative by forming the discrete adjoint equations from the implicit trapezoidal integration scheme, integrating them backward in time to compute the adjoint variables λ, and then using those λ values to efficiently evaluate the gradient of the final-state constraint residuals with respect to the control spline parameters and final time.

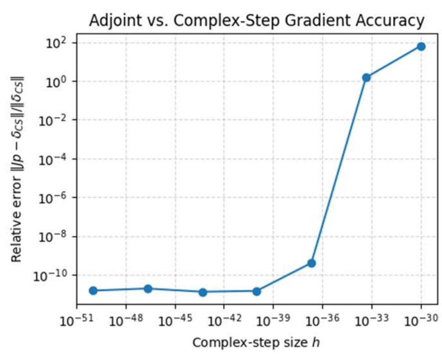

Figure 8: Adjoint vs Complex-Step Accuracy for AOA initialization as a sine wave starting at 0deg

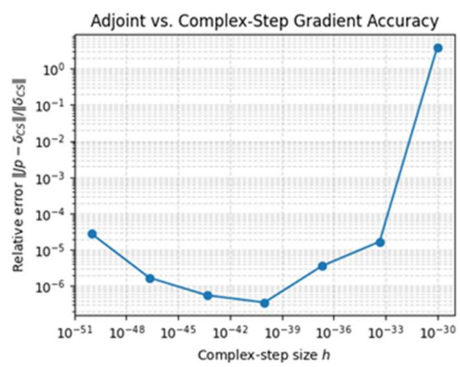

Figure 7: Adjoint vs Complex-Step Accuracy for AOA initialization as a sine wave starting at 2deg

B. Full-Space Method and Jacobian Accuracy

The full-space method enforces dynamics and boundary conditions through a full residual vector. Its Jacobian is computed using complex-step differentiation: each state and control variable was perturbed by a tiny imaginary increment, the full residual vector was evaluated, and the resulting imaginary parts, divided by the step size, directly yielded the Jacobian columns with machine-precision accuracy. Validation via Finite-difference comparison tests confirmed relative errors below 1e-9.

In the full-space formulation, both the state variables and their time derivatives are treated as independent variables. The optimization variables include the state vector $q(t) = [v, \gamma, h, r, m]$, control $\alpha(t)$, state derivatives $\dot{x}(t)$, and final time tf.

The objective is:

$$\min_{\{\alpha(t), q(t), \dot{x}(t), tf\}} tf$$

Subject to:

- Dynamic constraints: $R(t) = \dot{x} - f(q, \alpha) = 0$

- Boundary conditions at initial and final states
- Bounds on α(t) ∈ [−20°, 20°]

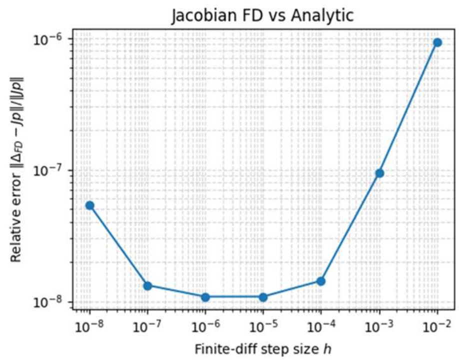

Figure 9: Full-Space Jacobian Accuracy

IV. Optimization Challenges and Modifications

Reduced-space trajectories occasionally diverged due to integration instability and poor initial guesses. These were mitigated by: (1) sinusoidal and shaped α initialization, (2) altitude, velocity and FPA clamping, and (3) relaxed Newton updates. Full-space convergence was more robust but sensitive to residual scaling and target weights. Optimization convergence took many iterations and required widening of the acceptable tolerances for final states. Additionally, for the Jacobian checks, the level of accuracy was often determined by the feasibility of the points you choose to test with (See Figures 7 and 8). The further away from feasible points you got, the less accurate or even demonstrably poor accuracy was found between Jacobians.

V. Key Results and Observations

Final-time results across both methods converged to ~350 seconds. Reduced-space runs were slower and harder to stabilize. Full-space runs had better constraint satisfaction but required more memory and tighter Jacobian accuracy. The success of the Reduced-space model greatly depended on a good guess provided at the beginning. I found that a smooth sine wave did the best starting at around 2 degrees positive AOA. I found that I needed to increase the number of control points in the Reduced-model to allow the optimizer to have more flexibility moving the AOA control to find more optimal or feasible conditions. The Full-space model ran well with many of the inputs I provided. Correctly choosing scaling parameters was the larger design challenge for that model. An additional challenge was in adjusting the target tolerances to allow the optimizer to find a converged solution that met the constraint criteria well. Additionally, I found that using warm-restarts helped improve progress greatly for both models. The Full-Space model optimal AOA Bang-bang control maneuvering is likely not physically reasonable for most aircraft. But it did provide a clarifying contrast to the Reduced-space model.

References

[1] A. E. a. D. W. F. Bryson, "A Steepest-Ascent Method for Solving Optimum Programming Problems," Journal of Applied Mechanics, vol. 29, no. 2, p. 247–257, 1962.

[2] J. T. Betts, Practical Methods for Optimal Control and Estimation Using Nonlinear Programming, Philadel. phia: Society for Industrial and Applied Mathematics, 2010.

Appendix

Source material figures for aerodynamic data:

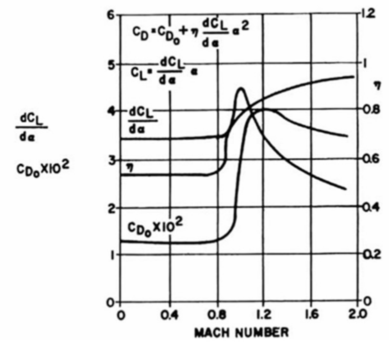

Figure 10: Original Aero Model Data [1]

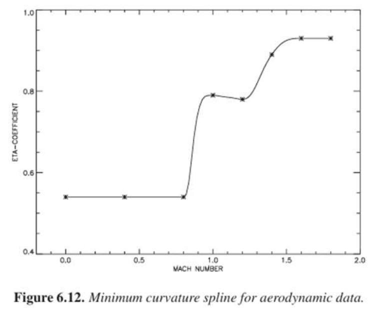

Figure 11: Minimum curvature spline for Eta [2]

Delaunay-Based Linear Interpolation (Thrust Model)

To model thrust as a function of Mach number and altitude, a piecewise planar interpolation was used based on a Delaunay triangulation of the 2D input space. Within each triangle (simplex), a local plane is fit using least squares:

$$f(M, h) = a*M + b*h + c$$

To compute the coefficients a, b, and c, a system of equations is solved for each triangle:

$$[A] \cdot [a\ b\ c]^T = [T1\ T2\ T3]^T$$

To support complex-step derivatives, the local gradient $grad\_f = [a, b]$ is used:

$$Im(f(x + ihp)) \approx h \cdot grad\_f \cdot p$$

Cubic Spline Interpolation (1D Models)

Used for density, speed of sound, and parts of the aerodynamic models. The cubic spline between points x_i and x_{i+1} is defined as:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

Cubic splines ensure continuity in value, slope, and curvature, with natural boundary conditions.Surrogate models are constructed to handle complex-valued inputs. For a function f(x), the directional derivative is computed using:

$$f'(x) \approx Im(f(x + ih)) / h$$

This allows gradient computation accurate to machine precision without subtractive cancellation errors.

Piecewise Linear + Spline Surrogates

Used for lift curve slope ($C_{L\alpha}$), zero-lift drag coefficient (C_{D0}), and efficiency factor (η). Below a breakpoint x_b, linear interpolation is used. Above x_b, a cubic spline is employed.

$$f(x) = \begin{cases} \text{linear interpolation, for } x \leq x_b \\ \text{spline}(x), \text{ for } x > x_b \end{cases}$$

Table 2: Summary of Surrogate Model Features

Model	Input(s)	Method	Complex-Step Compatible
Thrust	Mach, Altitude	Delaunay planar interpolation	Yes
C_Lα	Mach	Piecewise (linear + spline)	Yes
C_D0	Mach	Piecewise (linear + spline)	Yes
Density (ρ)	Altitude	Cubic spline	Yes
Speed of Sound (a)	Altitude	Cubic spline	Yes
Efficiency (η)	Mach	Piecewise (linear + spline)	Yes

Final Trajectory Plots for Converged Solutions

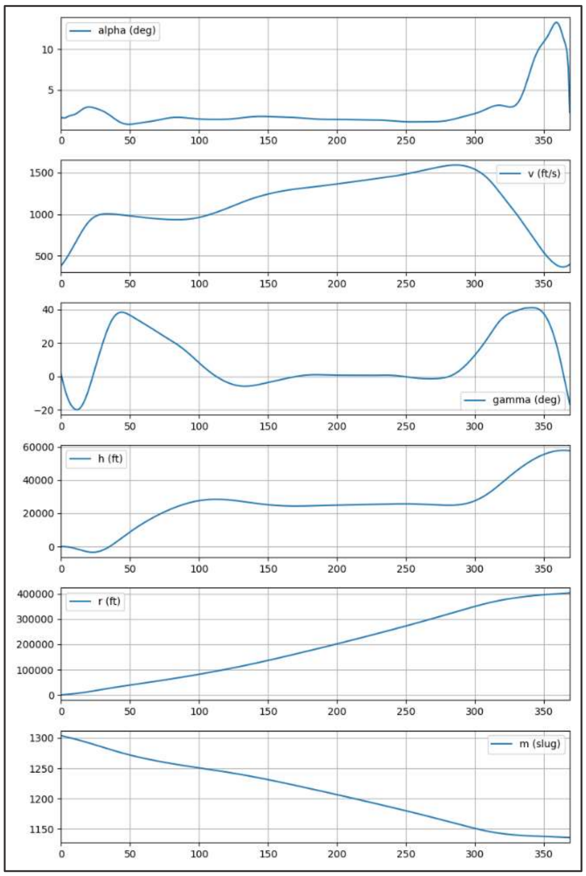

Figure 12: Reduced-Space Final Optimized Trajectory

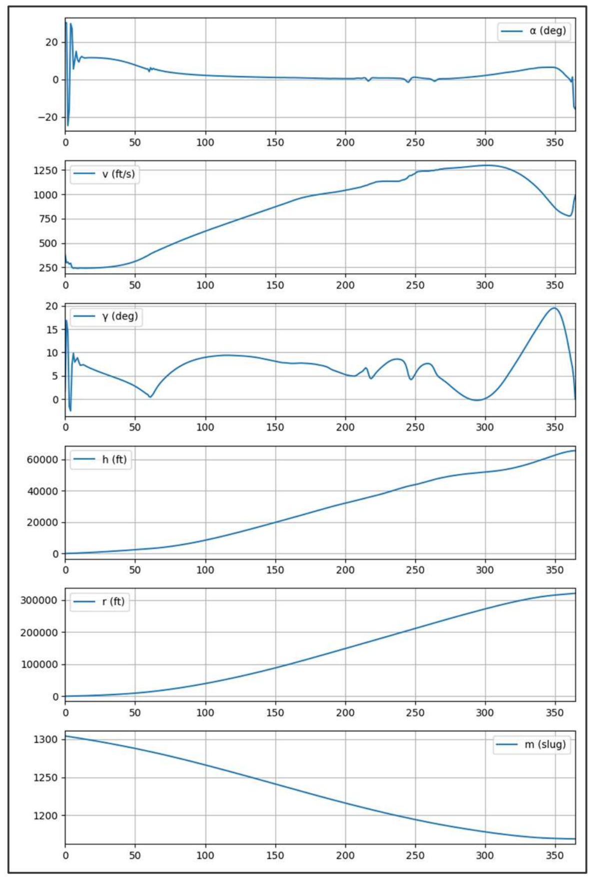

Figure 13: Full-Space Final Optimized Trajectory

Please see additional plots demonstrating optimization results for 3 different initial AOA (shaped, flat 0, flat negative 2, and random) guesses in attached Appendix B PDF.