
```
clear
clc
close all

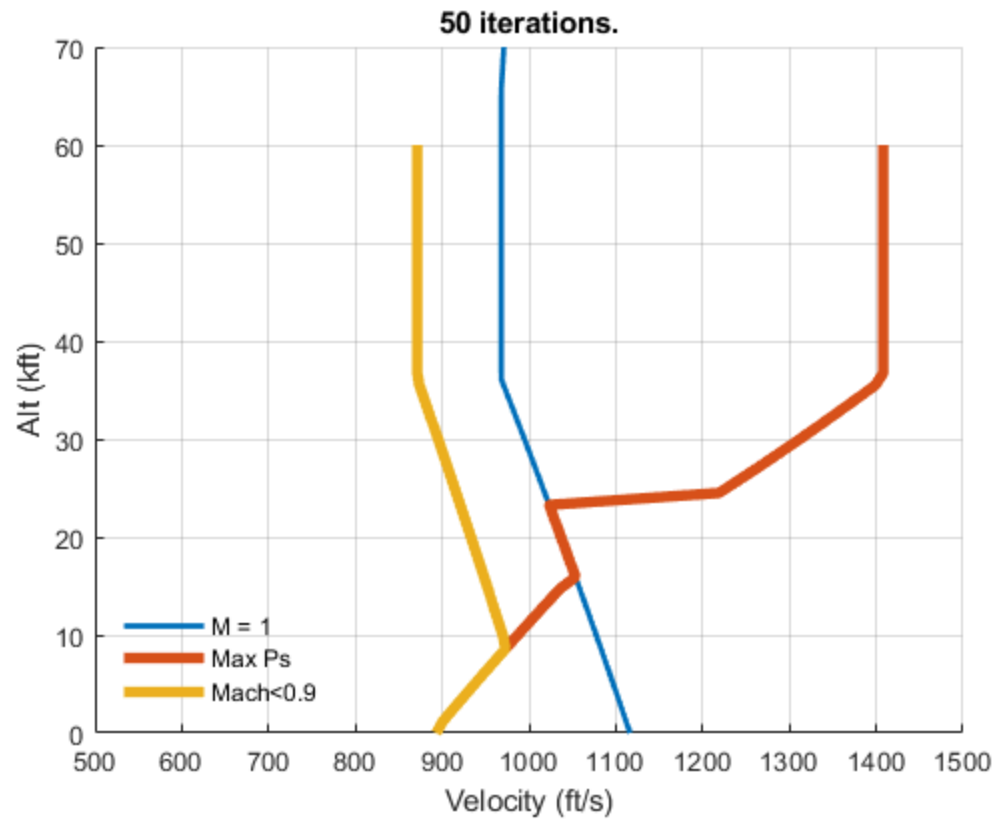
addpath("../")% Not needed if +PropPrelib folder is in your current
path.

import PropPrelib.*

units BE;
enginemodel LBTF;
dragmodel FutureFighter;
atmodel Standard;

h = 60000;
WL = 64;
TL = 1.25;
TR = 1.07;
h_TO = 0;
Beta_TO = 0.97;
AB = 0;
n = 1;

hold on
axis([500, 1500, 0, 70])
grid on
title(sprintf('%d iterations.', 50))
ylabel('Alt (kft)')
xlabel('Velocity (ft/s)')
H = linspace(0,70E3);
[~, V, ~] = atmos(H);
plot(V, H/1E3,'LineWidth',2)
[V, H] = MinTimeClimbSch(h, TR, TL, WL, Beta_TO, h_TO, n, AB, 50);
h_kft = H(1:2:end)'./1E3;
min_t_V = V(1:2:end)';
plot(V, H/1E3,'LineWidth',4)
V = MinTimeClimbSch(h, TR, TL, WL, Beta_TO, h_TO, n, AB, 50, 0.9);
less_M_09_V = V(1:2:end)';
plot(V, H/1E3,'LineWidth',4)
legend('M = 1', 'Max
Ps', 'Mach<0.9','Location', 'SouthWest', 'color','none','color','none','Box', 'of
drawnow
```



Table

```
disp(table(h_kft,min_t_V,less_M_09_V))
```

<i>h_kft</i>	<i>min_t_V</i>	<i>less_M_09_V</i>
0	893.16	893.16
2.449	912.7	912.7
4.898	936.43	936.43
7.3469	960.53	960.53
9.7959	985.05	970.38
12.245	1010	961.58
14.694	1035.5	952.7
17.143	1048.6	943.73
19.592	1038.5	934.68
22.041	1028.4	925.54
24.49	1218.9	916.31
26.939	1261.2	906.99
29.388	1302.1	897.57
31.837	1341.8	888.05
34.286	1380.4	878.42
36.735	1408.1	871.27
39.184	1408.1	871.27
41.633	1408.1	871.27
44.082	1408.1	871.27

46.531	1408.1	871.27
48.98	1408.1	871.27
51.429	1408.1	871.27
53.878	1408.1	871.27
56.327	1408.1	871.27
58.776	1408.1	871.27

Functions Used

```
function [V, H, Ps, M] = MinTimeClimbSch(h, TR, TL, WL, Beta_TO, h_TO,
n, AB, inc, M_Max)
    %MINTIMECLIMBSCH
    import PropPrelib.*
    persistent memMinTimeClimbSch
    if isempty(memMinTimeClimbSch)
        % Useful when running comparisons between mach limited and
        unrestriced velocities
        % Dont have to recalculate Ps/V relations, since this func can
        take like 30 seconds
        % per run
        memMinTimeClimbSch = memoize(@MinTimeClimbSchIMPL);
    end
    if nargin == 8
        inc = 5;
    end
    H = linspace(h_TO, h, inc);

    [V, Ps] = memMinTimeClimbSch(H, TR, TL, WL, Beta_TO, n, AB);

    [~, a] = atmos(H);
    M = V./a;
    if nargin == 10
        M(M>M_Max) = M_Max;
        V = M.*a;
    end
end

function [V, Ps] = MinTimeClimbSchIMPL(H, TR, TL, WL, Beta_TO, n, AB)
    import PropPrelib.*
    [V, Ps] = vecfun(@VmaxPs, H, TR, TL, WL, Beta_TO, n, AB);
end

function [V, Ps] = VmaxPs(h,TR, TL, WL, Beta_TO, n, AB)
    %[V, Ps] = VMAXPS (h, TR, TL, WL, beta, n, AB)
    %Given altitude, Throttle Ratio, Thrust Loading, Wing Loading,
    %beta, n, and AB settting, calculate velocity for maxexcess power.
    %Requires dragmodel set.
    %Requires enginemodel set.
    %Requires units set.
```

```

%Requires atmodel set.
opts = optimoptions(@fmincon,'Algorithm','sqp');
problem = createOptimProblem('fmincon','objective',...
    @negPs,'x0',1000,'lb',200,'ub',2000,'options',opts);
ms = MultiStart('Display','off');
[V,Ps] = run(ms,problem,8);
Ps = Ps.*-1;

function nPs = negPs(V)
    import PropPrelib.*
    nPs = 0-PsFVh(V, h, TR, TL, WL, Beta_TO, n, AB);
end
end

function Ps = PsFVh(V, h, TR, TL, WL, beta, n, AB)
%Ps = PSFVH (V, h, TR, TL, WL, beta, n, AB)
%Given Velocity, altitude, Throttle Ratio, Thrust Loading, Wing
Loading,
%beta, n, and AB settting, calculate excess power.
%Requires dragmodel set.
%Requires enginemodel set.
%Requires units set.
%Requires atmodel set.
    import PropPrelib.*
    [~, a, P, ~, theta, delta, ~] = atmos(h);
    M = V./a;
    [theta_0, delta_0] = adjust_atmos(theta, delta, M);
    alpha = thrustLapse(theta_0, delta_0, TR, 'AB', AB);
    q = dynamic_pressure(P, M);
    [K1, CD0, K2] = drag_constants(M);
    Ps = V*(alpha./beta.*TL - K1.*n.^2*beta./q.*WL - K2.*n - CD0./
((beta./q).*WL));
end

```

Published with MATLAB® R2019a