Table of Contents

```
function latticeout
```

Optima

This script outputs a fairly large lattice plot comparing V_cruise, Range and L/D max. The output axis for the lattice plot will be Wing loading, WS and power req, P

```
% Alex M Granata
% 03 OCT 2015

close all; clear; clc
```

Initial Assumptions and Requirments of flight

```
% Payload
%     19 Passengers w/ Cargo @ 225 lbs each, plus crew and attendent
Wfix=(19+2+1)*225;  % lbm
% Initial Performance Constants
sfc=0.55;    % specifc fuel consumption, lb_fuel per hour / lb_thrust
E=0;     % i.e. 25 min loiter time
AR=10;      % Aspect Ratio
Cd0=0.022;  %
e=0.8;      % Oswald Efficency
% Cruise Enviromental Conditions
p_c=1.267e-3;  % slugs per ft^3
% Takeoff Distance
s_T=3000;   % ft
% WS Restraint Constants
Cl_max=1.7;  % Max C_L for takeoff
V_stall=75*1.688; % Stall speed in ft/s, Sea Level, max 75 knots
% Convience Conditions
K=1/(e*pi*AR);
V_md=@(ws,p) sqrt(2*ws/(p*sqrt(Cd0/K)));
V_mp=@(ws,p) sqrt(2*ws/p)*(K/3*Cd0)^(1/4);
p_sl=2.3769e-3; %slugs/ft^3, sea level density
% Service Ceiling
p_sc=.958e-3;   % slugs per ft^3
```

Domains of independent variables

```
Rl=5;  Rd=linspace(800,1200,Rl)*1.151;  %miles
Vl=6;  Vd=linspace(250,300,Vl)*1.466667;  %ft/sec
LDl=2;  LDd=linspace(18,22,LDl);  % Keep resolution of LD <5
WTOl=4; % Number of isolines for the WTO contours
```

Numerical Calculations

For loops to calculate WS, TW and P for independent Variables

```
tic
optimzd=zeros(Rl,Vl,LDl,3);
poolobj = gcp('nocreate'); % If no pool, do not create new one.
if isempty(poolobj)
    parpool('local')
end

for it_LD=1:LDl
    LD=LDd(it_LD);
    for it_R=1:Rl
        R=Rd(it_R);
        for it_V=1:Vl
            V_c=Vd(it_V);

            % Fuel Weight Fraction
            % Fuel ratios
            Wr=exp(-R*sfc/(V_c/1.46666667*(0.943*LD)));
            We=exp(-E*sfc/(LD));
            % Full ratio
            W1_6=0.97*0.985*Wr*1*We*0.985;
            Wf_Wto=1.06*(1-W1_6);
            % Empty Weight
            % From Trend line data, We=XX Wto ^ YY, XX=0.911, YY=0.947
            Wept=@(Wt) 0.911*Wt^0.947;
            % Takeoff Weight Estimation
            % Compare We_est to Wto_est - Wfix - Wfuel
            Wto=fsolve(@(wt) wt*(1-Wf_Wto)-Wfix-
Wept(wt),30e3,optimoptions('fsolve','Display','off'));    % Estimate
 W_TO

            % Wing-loading min to meet C_L_max and V_stall assumptions
            WS_s=0.5*p_c*V_stall^2*Cl_max;
            % Min loading for to meet Range requirement
            WS_r=fsolve(@(x) Wf_Wto*Wto*1.07/
sfc*(x/(p_c))^0.5*(AR*e)^(1/4)/Cd0^(3/4)*(1/Wto)-
R,10,optimoptions('fsolve','Display','off'));
            % Limit on W/S for Landing Distance within required
 distance, 3 degree
            % angle of approach
            WS_l=fsolve(@(ws) s_T-(79.4*ws/(1*Cl_max)+50/tand(3)),...
                50,optimoptions('fsolve','display','off'));
            % Drag for Power
            D=@(ws,V,p,dhdt,n) (Wto*dhdt)/V + (Cd0*V^2*Wto*p)/(2*ws) +
 (2*K*Wto*n^2*ws)/(V^2*p);

            % Calculations for Power
            % Straight, Level Flight
            Preq_cruise=@(ws) D(ws,V_c,p_c,0,1)*V_c/(p_c/p_sl);
            % Service Ceiling
            Preq_serv=@(ws)
 (D(ws,V_mp(ws,p_sc),p_sc,100/60,1).*V_mp(ws,p_sc)/(p_sc/p_sl));
            % Cruise Ceiling
            Preq_cc=@(ws)
 D(ws,V_mp(ws,p_c),p_c,300/60,1).*V_mp(ws,p_c)/(p_c/p_sl);
            % 2.5g Maneuer at Sea Level
            Preq_man=@(ws) D(ws,V_c,p_sl,0,2.5)*V_c/(p_sl/p_sl);

            % Takeoff Power, from Takeoff T/W
            VTO=V_stall*1.2;
            n=6; d=5;
            T0=@(p) 0.25*(p./n.*d).^(2/3);
            T0t=@(p) T0(p).*n;
            a=@(p) ((p/VTO)-T0t(p))/(VTO)^2;
            A=@(p) 32.2*(T0t(p)/15000-0.04);
            mu=0.04; % Firm Turf
            Cdg=0.0527; %Ground roll drag
            CLg=1.7; % max ground lift
            B=@(p,ws) 32.2/Wto*(0.5*p_sl*(Wto/ws)*(Cdg-mu*CLg)+a(p));
            s=@(p,ws) 1./(2*B(p,ws)).*log(A(p)./(A(p)-
B(p,ws)*(VTO)^2));
            Preq_TOop=@(ws) fsolve(@(p)
 s(p,ws)-2000,600*550,optimoptions('fsolve','Display','off'));

            mat=[{Preq_cruise};{Preq_cc};{Preq_man};{Preq_serv};
{Preq_TOop}];
            [wsx,py]=fminbnd(@(ws) ...
                max([mat{1}(ws),mat{2}(ws),mat{3}(ws),mat{4}
(ws),mat{5}(ws)]),...
                max(WS_r,WS_s),WS_l);
            if numel(wsx>1)
                wsx=wsx(1);
            end
            optm=[wsx,py/550];    % in lbs/ft^2 and hp

            optimzd(it_R,it_V,it_LD,:)=[optm,Wto]; % Save Optimized
 Data
        end
    end
end
toc
```

Figure Setup

Base figure will be used to output the optimized Ws/Power, and control the other two design space and optimizer plots

```
r=groot;
if size(r.MonitorPositions,1)>1
    bspos=r.MonitorPositions(1,:);
    f_detpos=r.MonitorPositions(2,:);
else
    bspos=r.MonitorPositions(1,:).*[1 1 1 0.5];

 f_detpos=[r.MonitorPositions(1,1),r.MonitorPositions(1,4)/2,r.MonitorPositions(1,
 0.5]];
end

bs=figure('Name','Optimized Power and Wing Loading',...
    'NumberTitle','off',...
    'DockControls','off',...
    'MenuBar','none',...
    'Units','pixels',...     'Resize','off',...
    'Position',bspos,...
    'deletefcn','close all; clear; clc;');

% Setup the axes for
abse=axes('Parent',bs);
abse.XLimMode='manual';
abse.Title.String='Power Requirments and Wing Loading';
abse.XLabel.String='Wing Loading W/S, lbs/ft^2 by LD_{max}';
abse.YLabel.String='Power_{Required}, hp';
abse.Color='none';
hold(abse,'on')

% Details figure will be a split plot graphic the details of the
 selected
% point int he bas figure
f_det=figure('Name','Point Specifc Details',...
    'NumberTitle','off',...
    'DockControls','off',...
    'MenuBar','none',...
    'Units','pixels',...'Resize','off',...
    'Position',f_detpos,...
    'deletefcn','close all; clear; clc;');
aTW=axes('Parent',f_det);
aTW.XLimMode='manual';
subplot(1,2,1,aTW);
aTW.Title.String='Constraints on Wing Loading and Specifc Thrust';
aTW.XLabel.String='Wing Loading W/S';
aTW.YLabel.String='Specfic Thrust: T/W';
hold(aTW,'on')
aPW=subplot(1,2,2);
aPW.Title.String='Operational Power Curves';
aPW.XLabel.String='Wing Loading W/S';
aPW.YLabel.String='Power_{Required}, hp';
hold(aPW,'on')

clear r bspos f_detpos
```

Setup Xaxis for Lattice Plot

```
abse.Units='pixels';
szax=abse.Position;
abse.Units='normalized';
catlim=round(szax(3)/LDl/50-2,0);
rnge=[min(min(min(optimzd(:,:,:,1)))),max(max(max(optimzd(:,:,:,1))))];
magni=round(log10(diff(rnge)))-1;
% Determined magnitude of steps
optmsmag=[1,2,5];
rangeoptions=optmsmag'*10^magni*catlim;
[~,ir]=min(abs(rangeoptions-diff(rnge)));
stepsz=optmsmag(ir)*10^(magni); %determine size of nominal steps
lbs=floor(rnge(1)/stepsz)*stepsz:stepsz:ceil(rnge(2)/
stepsz)*stepsz; %Actual steps
x_ticklabel={' ',lbs};
x_tickvalue=1:((length(lbs)+1)*LDl);
abse.XTickLabel=x_ticklabel;
abse.XTick=x_tickvalue;
abse.XLim=[1 ((length(lbs)+1)*LDl)];

clear szax catlim rnge magni optmsmag rangeoptions ir stepsz
```

Plot the V/R Optimized Data

Split Lattice plot in order to display data

```
figure(bs);
axes(abse);
xint=zeros(Rl,Vl,LDl);
yint=zeros(Rl,Vl,LDl);
% Plot Each Carpet by successive LD settings
for a=1:LDl
    %     Velocity Dependant
    for b=1:Vl

 xint(:,b,a)=(a-1)*(length(lbs)+1)+1+interp1(lbs,1:length(lbs),optimzd(:,b,a,1),'p
        plot(abse,xint(:,b,a), optimzd(:,b,a,2),'Color',0.8*[1 1 1])
        if Vl*LDl>20
            if b==1 || b==Vl || b==round(Vl/2)
                text(xint(1,b,a),...
                    optimzd(1,b,a,2),...
                    sprintf('%.4g mph',Vd(b)/1.46666667),...

  'Verticalalignment','top','horizontalAlignment','left')
            else
                continue
            end
        else
            text(xint(1,b,a),...
                optimzd(1,b,a,2),...
                sprintf('%.4g mph',Vd(b)/1.46666667),...

  'Verticalalignment','top','horizontalAlignment','left')
        end
    end

    %     Range Dependant
    for c=1:Rl

 yint(c,:,a)=(a-1)*(length(lbs)+1)+1+interp1(lbs,1:length(lbs),optimzd(c,:,a,1),'p
        plot(abse,yint(c,:,a),optimzd(c,:,a,2),'Color',0.8*[1 1 1])
        if Rl>5
            if c==1 || c==Rl || c==round(Rl/2)
                text(yint(c,1,a),...
                    optimzd(c,1,a,2),...
                    sprintf('%.4g nm',Rd(c)/1.151),...

  'Verticalalignment','middle','horizontalAlignment','right')
            else
                continue
            end
        else
            text(yint(c,1,a),...
                optimzd(c,1,a,2),...
                sprintf('%.4g nm',Rd(c)/1.151),...

  'Verticalalignment','middle','horizontalAlignment','right')
        end
    end
end

clear yint
```

LD Max Indicators

Connection Lines for LD_max boxes

```
ldconres=25;
for b=[1 Vl]
    for c=[1 Rl]
        tstr0(1:LDl)=optimzd(c,b,:,1); outr0(1:LDl)=optimzd(c,b,:,2);
        tstr1=((1:LDl)-1)*(length(lbs)+1)+1+...
            interp1(lbs,1:length(lbs),tstr0,'pchip');

 outr1=interp1(tstr1,outr0,linspace(tstr1(1),tstr1(end),ldconres),'pchip');

 plot(abse,linspace(tstr1(1),tstr1(end),ldconres),outr1,'Color',0.8*[1
 1 1]);
    end
end
```

```
% Text Label for LD_Max
for a=1:LDl
    text((a-1)*(length(lbs)+1)+round(length(lbs)/2+1),...
        abse.YLim(1)+0.93*diff(abse.YLim),...
        sprintf('LD_{max}: %.3g',LDd(a)),...
        'HorizontalAlignment','center','Color','r')
end

clear tstr0 tstr1 outr0 outr1
```

Gross Takeoff Weight Isolines

Draw isolines on each independante carpet plot of the W_TO for the V/R pairing. This plot is unique such that it all has to be done at once, so that a colorbar may be added to the side of the plot for indications

```
% First, define an axis that matches directly on top of abse
abs2=axes('Parent',bs);
abs2.Units='pixels';
abs2.XLimMode='manual';
abs2.Title.String=' ';
abs2.XLabel.String=' ';
abs2.YLabel.String=' ';
hold(abs2,'on')
abs2.XTickLabel=' ';
abs2.XTick=x_tickvalue;
abs2.XColor='none'; abs2.YColor='none';
abs2.XLimMode='manual'; abs2.XLim=abse.XLim;
abs2.YLimMode='manual'; abs2.YLim=abse.YLim;
abs2.ZLimMode='manual';
bs.Children=[bs.Children(2);bs.Children(1)];

% Grab C data from each WTO/LD system,
for a=1:LDl
    xint0=xint(:,:,a);
    optimzd0=optimzd(:,:,a,2);
    optimzd1=optimzd(:,:,a,3)/1000;
    contour(abs2,xint0,optimzd0,optimzd1,WTOl,'LineWidth',1.3);
end
clb=colorbar;
clb.Label.String='W_{gross takeoff}, 1,000 lbs';
abse.Units='pixels';
abse.Position=abs2.Position;
abse.Units='normalized'; abs2.Units='normalized';

clear xint optimzd0 optimzd1
```

Data Tip Details and Updater

Turn on Data Cursor mode for base figure, and assign the appropriate callback function to graph the details

```
bs.CurrentAxes=abse;
bsdc=datacursormode(bs);
```

```
bsdc.UpdateFcn=@drawdetails;
bsdc.Enable='on';

    function output_txt = drawdetails(~,evntobj)
        % Used to output custom datatip information, as well as update
 the detail
        % graphs that show specic information about the selected data.
        pos = get(evntobj,'Position');
        ldind=floor(pos(1)/(length(lbs)+1))+1;
        ldout=LDd(ldind);
        [ri,ci]=find(optimzd(:,:,ldind,2)==pos(2));
        WSout=optimzd(ri,ci,ldind,1);
        Wto_out=optimzd(ri,ci,ldind,3);
        posout=[interp1(1:length(lbs),lbs,pos(1)-floor(pos(1)/
(length(lbs)+1))*(length(lbs)+1)-1),...
            pos(2)];
        output_txt = {['LDmax: ',num2str(ldout,3)],...
            ['WS: ',num2str(WSout,4)],...
            ['P: ',num2str(pos(2),4)],...
            ['W_{takeoff}:  ',num2str(Wto_out)]};

        % Begin graphics
        [Rin,V_cin]=find(optimzd(:,:,floor(pos(1)/
(length(lbs)+1))+1,2)==pos(2));
        R=Rd(Rin);
        V_c=Vd(V_cin)/1.46666;
        LD=LDd(floor(pos(1)/(length(lbs)+1))+1);
        WS_r=[]; WS_s=[]; WS_l=[]; TW_c=[]; TW_cruise=[]; TW_serv=[];
 TW_cc=[];
        TW_man=[]; ym=[]; xm=[]; WSD=[]; s_tm=[];
        Preq_TO=[]; Pr=[];

        ymabs=abse.YLim;
        keyboard
        cnstr_n
        power_cnstr_n
    end

end
```

Published with MATLAB® R2015a