Oblique Cutting

Ratnangshu Das, 17807558

January 27, 2021

Question 1

One useful way of validating the different models and understanding their importance is to compare their results with practical or experimental data. This is what we will do in this question. The objective in this question would be to analyse the cutting forces in oblique cutting for a given shear yield stress of the material and friction coefficient.We assume that the shear yield stress in the shear plane, and friction coefficient on the rake face are constant.

Consider oblique cutting of a material that has a yield strength of 750 MPA with a tool that has a normal rake angle of $\alpha_n = 20°$. Assume the friction angle β_a =34.6°. The uncut chip thickness h = 0.5 mm, the cut chip thickness h_c = 1.1 mm, and the width of cut b = 6.25 mm. The data is in accordance with the experimental data published by Lin and Oxley (1972). And as for the angle of obliquity, i is concerned, we will vary it from 0° to 30°.

Now, there are 5 unknown oblique cutting parameters, which define the direction of resulting force:
ϕ_n: normal shear angle,
ϕ_i: oblique shear angle,
θ_n: normal angle of resultant cutting force direction,
θ_i: oblique angle of resultant cutting force direction, and
η: chip flow angle.

From the oblique cutting geometry, we obtain the following 3 equations:

$$\sin(\theta_i) = \sin(\beta_a)\sin(\eta) \tag{1}$$
$$\tan(\theta_n + \alpha_n) = \tan(\beta_a)\cos(\eta) \tag{2}$$
$$\tan(\eta) = \frac{\tan(i)\cos(\phi_n - \alpha_n) - \cos(\alpha_n)\tan(\phi_i)}{\sin(\phi_n)} \tag{3}$$

We need additional two expressions to solve the 5 unknown angles. One way to do that is to extend Merchant's **minimum energy principle** from orthogonal cutting to oblique cutting. The minimum energy principle requires that the cutting power must be minimum for a unique shear angle solution. Thus, we have 2 more equations:

$$\frac{\partial P_t'}{\partial \phi_n} = 0; \tag{4}$$
$$\frac{\partial P_t'}{\partial \phi_i} = 0; \tag{5}$$

where the normalized power is: $P_t' = \frac{P_t c}{V \tau_s b h} = \frac{\cos(\theta_n) + \tan(\theta_i)\tan(i)}{[\cos(\theta_n + \phi_n)\cos(\phi_i) + \tan(\theta_i)\sin(\phi_i)]\sin(\phi_n)}$

1. Find the five unknown oblique angles using the numerical iteration technique and plot them as a function of i.

Now, another way to solve this is instead of adding 2 more equations, we introduce some assumptions and reduce the no. of unknown parameters to 3. This is the **empirical approach of Armarego and**

Whitfield. The equations 1 to 3 transform to:

$$\tan(\phi_n + \beta_a) = \frac{\cos(\alpha_n)\tan(i)}{\tan(\eta) - \sin(\alpha_n)\tan(i)} \tag{6}$$

$$\tan(_n) = \tan(\beta_a)\cos(\eta) \tag{7}$$

$$\phi_n = \tan^{-1}\left(\frac{r_c(\cos(\eta)/\cos(i))\cos(\alpha_n)}{1 - r_c(\cos(\eta)/\cos(i))\sin(\alpha_n)}\right) \tag{8}$$

where $\beta_n = \theta_n + \alpha_n$, and $r_c = \frac{h}{h_c}$ is the chip thickness ratio.

2. Solve the above three equations (Eq. 6-8) numerically to find the three unknown angles: β_n, ϕ_n and η and plot them as a function of i.

3. Adopt Stabler's empirical chip rule for simplification, i.e. $\eta = i$ and calculate β_n, ϕ_n and η using Eq. (6-8) and plot them as a function of i.

[resume] Finally we will plot only the F_r component of the force as its data is present in Lin and Oxley (1978) and we can compare. So,

1. Plot F_r as a function of i for all the three methods.

Plot all of them onto same respective graphs.

Now plot the experimental data points from Lin and Oxley (1978). As there are data for multiple cutting speeds, choose cutting speeds: 400 ft/min and 600 ft/min.

i (degrees)	**U (ft/min)**	**U (m/min)**	η **(degrees)**	β_a **(degrees)**	ϕ **(degrees)**	**lbf**			**N**		
						F_t	F_f	F_r	F_t	F_f	F_r
0	25	7.62	0	29.45	28.9	613	102	0	2726.759	453.7184	0
	200	60.96	0	32.67	27.2	667	150	0	2966.963	667.233	0
	400	121.92	0	33.82	26.3	683	168	0	3038.134	747.301	0
	600	182.88	0	32.91	26.1	663	152	0	2949.17	676.1294	0
	800	243.84	0	27.97	26.9	593	83	0	2637.794	369.2023	0
	1000	304.8	0	26.59	19.5	692	80	0	3078.168	355.8576	0
10	25	7.62	10	26.54	30.2	597	69	70	2655.587	306.9272	311.3754
	200	60.96	9.6	31.22	27.9	668	133	74	2971.411	591.6133	329.1683
	400	121.92	9.5	32.09	27.2	674	145	75	2998.1	644.9919	333.6165
	600	182.88	9.8	33.52	25.8	692	167	76	3078.168	742.8527	338.0647
	800	243.84	9.8	27.63	26.9	595	80	66	2646.691	355.8576	293.5825
	1000	304.8	10.4	25.37	20	691	65	65	3073.72	289.1343	289.1343
20	25	7.62	19.3	25.83	30.5	615	64	143	2735.655	284.6861	636.0955
	200	60.96	20	29.45	28.6	669	113	147	2975.859	502.6489	653.8883
	400	121.92	20	31.47	27.4	690	142	150	3069.272	631.6472	667.233
	600	182.88	19.7	33.15	25.8	714	169	152	3176.029	751.7492	676.1294
	800	243.84	20.5	27.32	27	614	80	135	2731.207	355.8576	600.5097
	1000	304.8	21.8	24.22	20.2	691	51	124	3073.72	226.8592	551.5793
30	25	7.62	29.6	24.08	31.2	620	46	216	2757.896	204.6181	960.8155
	200	60.96	28.6	27.03	29.7	670	85	218	2980.307	378.0987	969.712
	400	121.92	28.7	32.03	27	726	158	226	3229.408	702.8188	1005.298
	600	182.88	29.3	31.6	26.4	735	154	228	3269.442	685.0259	1014.194
	800	243.84	30.4	26.78	27	659	80	205	2931.377	355.8576	911.8851
	1000	304.8	30.8	23.16	20.4	695	39	210	3091.513	173.4806	934.1262

Table 1: Experimental Data from Lin and Oxley (1978)

Finally, we can see, apart from some acceptable engineering errors, the models can almost trace the experimental data.

Solution

We iterate through the values of i from 1 to 45 and with the rest of the known angles, we proceed through the calculations like this.

First, we solve via numerical iteration using the minimum energy principle. The mechanics of oblique cutting are defined by five equations. Three of them are obtained from the oblique cutting geometry expression, and the remaining two are derived by applying the Minimum Energy Principle, extended from Merchant's proposal for orthogonal cutting.

Next we follow the Armarego's method and by assuming:

1. Shear velocity is collinear with shear force from Stabler's work
2. Chip length ratio in oblique is same as in orthogonal from experiments

we reduce the no. of unknowns to 3. We now followed two methods:

- solved the three equations and three unknowns via numerical iteration and
- simply assuming Stabler's criterion, solve them analytically.

Then we calculate the forces in each of the case and finally plot them onto respected graphs for comparison.

This is the well commented Matlab code for all of these:

```
clc;
clear;

%% Values according to Lin & Oxley (1972)
alpha_n = 20*(pi/180); % normal rake angle
beta_a = 32.57375*(pi/180); % friction angle, taken average
h = 0.5; % given depth of cut
hc = 1.1; % assumed cut chip thickness
rc = h/hc; % chip thickness ratio
v = 0.1; % interpolation ratio selected within the range 0 < v < 1

%% Initializing arrays
theta_n_val_arr = zeros(1,45);
beta_n_a_arr = zeros(1,45);
beta_n_s_arr = zeros(1,45);
theta_i_val_arr = zeros(1,45);
phi_n_val_arr = zeros(1,45);
phi_n_a_arr = zeros(1,45);
phi_n_s_arr = zeros(1,45);
phi_i_val_arr = zeros(1,45);
eta_temp_arr = zeros(1,45);
eta_temp_a_arr = zeros(1,45);
eta_s_arr = zeros(1,45);
Ft_arr = zeros(1,45);
Ff_arr = zeros(1,45);
Fr_arr = zeros(1,45);
Ft_a_arr = zeros(1,45);
Ff_a_arr = zeros(1,45);
Fr_a_arr = zeros(1,45);
Ft_s_arr = zeros(1,45);
Ff_s_arr = zeros(1,45);
Fr_s_arr = zeros(1,45);

%% Start of loop
for ang = 1:1:45
    i = ang*(pi/180); % inclination angle
    eta_temp = i; % chip flow angle for "minimum energy" calculation part
    eta_temp_a = i; % chip flow angle for "armarego" calculation part
```

```
for k = 0:30 % the iteration loop typically converges by 30 iterations;
             %  can be replaced by a while loop

    %% Minimum Energy Principle
    % Force Relation
    eta_val = eta_temp;
    theta_i_val = asin(sin(beta_a)*sin(eta_val));
    theta_n_val_temp = atan(cos(eta_val)*tan(beta_a)) - alpha_n;
    theta_n_val = atan(sin(theta_n_val_temp)/cos(theta_n_val_temp));

    % Shear angle prediction
    syms phi_n phi_i
    Pt_p = (cos(theta_n_val) + tan(theta_i_val)*tan(i))/((cos(theta_n_val
            + phi_n)*cos(phi_i) + tan(theta_i_val)*sin(phi_i))*sin(phi_n));
    eqn4 = diff(Pt_p, phi_n) == 0;
    eqn5 = diff(Pt_p, phi_i) == 0;
    SSA = vpasolve([eqn4, eqn5], [phi_i, phi_n], [-pi/2 pi/2; -pi/2 pi/2]);
    %SSA = vpasolve([eqn4, eqn5], [phi_i, phi_n], [0; 40*pi/180]);
    phi_i_val = double(SSA.phi_i);
    phi_n_val = double(SSA.phi_n);

    % Velocity relation
    eta_new = atan(double(subs((tan(i)*cos(phi_n_val - alpha_n) -
                cos(alpha_n)*tan(phi_i_val))/(sin(phi_n_val)))));
    eta_temp = v*eta_temp + (1-v)*eta_new;

    % Power
    Pt_p = (cos(theta_n_val) + tan(theta_i_val)*tan(i))/((cos(theta_n_val + phi_n_val)*
            cos(phi_i_val) + tan(theta_i_val)*sin(phi_i_val))*sin(phi_n_val));
    % plot(k,Pt_p,'bo'); hold on; % this could be plotted for separate
    % cases to see how power converges with the iterations

    %% Armarego
    eta_a = eta_temp_a;
    beta_n_a = atan(tan(beta_a)*cos(eta_a));
    phi_n_a = atan((rc*(cos(eta_a)/cos(i))*cos(alpha_n))/(1-rc*(cos(eta_a)/cos(i))*sin(alpha_n)));
    if k ==1 % for simplified armarego case, which directly assumes eta = i;
        eta_s = eta_temp_a;
        phi_n_s = phi_n_a;
        beta_n_s = beta_n_a;
    end
    eta_temp_a = atan((cos(alpha_n)*tan(i))/(tan(phi_n_a + beta_n_a)) + (sin(alpha_n)*tan(i)));
end

%% Force Caluclations
% Minimum Energy
[Ft_arr(ang), Ff_arr(ang), Fr_arr(ang)] = calcForce(i, theta_i_val, theta_n_val, phi_i_val,
                                            phi_n_val, 0, alpha_n, eta_temp, 1);

% Armarego
[Ft_a_arr(ang), Ff_a_arr(ang), Fr_a_arr(ang)] = calcForce(i, 0, 0, 0, phi_n_a, beta_n_a,
                                                alpha_n, eta_temp_a, 0);

% simplified Armarego
```

```
    [Ft_s_arr(ang), Ff_s_arr(ang), Fr_s_arr(ang)] = calcForce(i, 0, beta_n_s-alpha_n, 0, phi_n_s,
                                                    beta_n_s, alpha_n, eta_s, 0);

    %% Storing values in arrays
    theta_n_val_arr(ang) = theta_n_val;
    beta_n_a_arr(ang) = beta_n_a;
    beta_n_s_arr(ang) = beta_n_s;
    theta_i_val_arr(ang) = theta_i_val;
    phi_n_val_arr(ang) = phi_n_val;
    phi_n_a_arr(ang) = phi_n_a;
    phi_n_s_arr(ang) = phi_n_s;
    phi_i_val_arr(ang) = phi_i_val;
    eta_temp_arr(ang) = eta_temp;
    eta_temp_a_arr(ang) = eta_temp_a;
    eta_s_arr(ang) = eta_s;
end

ang_arr = 1:1:45;
%% Plotting
subplot(3,3,1)
plot(ang_arr,theta_n_val_arr*(180/pi),'b'); hold on;
plot(ang_arr,(beta_n_a_arr - alpha_n)*(180/pi),'m'); hold on;
plot(ang_arr,(beta_n_s_arr - alpha_n)*(180/pi),'r'); hold on;
xlim([0 45])
ylim([-20 20])
xlabel('Inclination angle i (degrees)')
ylabel('\theta_n (degrees)')
grid on
grid minor

subplot(3,3,2)
plot(ang_arr,theta_i_val_arr*(180/pi),'b'); hold on;
xlim([0 45])
ylim([0 50])
xlabel('Inclination angle i (degrees)')
ylabel('\theta_i (degrees)')
grid on
grid minor

subplot(3,3,4)
plot(ang_arr,phi_n_val_arr*(180/pi),'b'); hold on;
plot(ang_arr,phi_n_a_arr*(180/pi),'m'); hold on;
plot(ang_arr,phi_n_s_arr*(180/pi),'r'); hold on;
xlim([0 45])
ylim([0 80])
xlabel('Inclination angle i (degrees)')
ylabel('\phi_n (degrees)')
grid on
grid minor

subplot(3,3,5)
plot(ang_arr,phi_i_val_arr*(180/pi),'b'); hold on;
xlim([0 45])
ylim([0 50])
```

```
xlabel('Inclination angle i (degrees)')
ylabel('\phi_i (degrees)')
grid on
grid minor

subplot(3,3,6)
plot(ang_arr,eta_temp_arr*(180/pi),'b'); hold on;
plot(ang_arr,eta_temp_a_arr*(180/pi),'m'); hold on;
plot(ang_arr,eta_s_arr*(180/pi),'r'); hold on; % armarego simplified
xlim([0 45])
ylim([0 80])
xlabel('Inclination angle i (degrees)')
ylabel('\eta (degrees)')
grid on
grid minor

subplot(3,3,7)
plot(ang_arr, Ft_arr/1000, 'b'); hold on;
plot(ang_arr, Ft_a_arr/1000, 'm'); hold on;
plot(ang_arr, Ft_s_arr/1000, 'r'); hold on;
xlim([0 45])
ylim([0 5])
xlabel('Inclination angle i (degrees)')
ylabel('F_{t} (kN)')
grid on
grid minor

subplot(3,3,8)
plot(ang_arr, Ff_arr/1000, 'b'); hold on;
plot(ang_arr, Ff_a_arr/1000, 'm'); hold on;
plot(ang_arr, Ff_s_arr/1000, 'r'); hold on;
xlim([0 45])
ylim([0 1])
xlabel('Inclination angle i (degrees)')
ylabel('F_{f} (kN)')
grid on
grid minor

subplot(3,3,9)
plot(ang_arr, Fr_arr/1000, 'b'); hold on;
plot(ang_arr, Fr_a_arr/1000, 'm'); hold on;
plot(ang_arr, Fr_s_arr/1000, 'r'); hold on;
xlim([0 45])
ylim([0 5])
xlabel('Inclination angle i (degrees)')
ylabel('F_{r} (kN)')
grid on
grid minor

% Plotting Experimental Values
T = readtable('Book1.xlsx'); % table containing experimental data from lin and oxley 1972
colors = [[0, 0.4470, 0.7410]; [0.8500, 0.3250, 0.0980]; [0.9290, 0.6940, 0.1250];
          [0.4940, 0.1840, 0.5560]; [0.4660, 0.6740, 0.1880]; [0.3010, 0.7450, 0.9330]];
irange = 0:10:30;
```

```
for sp_k = 3:4 % iterating for data of different cutting velocities U
    subplot(3,3,6) % plot eta data
    etarange = table2array(T(sp_k:7:sp_k+3*7,4));
    plot(irange,etarange,'*','color',colors(sp_k,:)); hold on;

    subplot(3,3,4) % plot phi_n data
    phi_n_range = table2array(T(sp_k:7:sp_k+3*7,6));
    plot(irange,phi_n_range,'*','color',colors(sp_k,:)); hold on;

    subplot(3,3,7) % plot F_t data
    F_t_range = table2array(T(sp_k:7:sp_k+3*7,10))/1000;
    plot(irange,F_t_range,'*','color',colors(sp_k,:)); hold on;

    subplot(3,3,8) % plot F_f data
    F_f_range = table2array(T(sp_k:7:sp_k+3*7,11))/1000;
    plot(irange,F_f_range,'*','color',colors(sp_k,:)); hold on;

    subplot(3,3,9) % plot F_r data
    F_r_range = table2array(T(sp_k:7:sp_k+3*7,12))/1000;
    plot(irange,F_r_range,'*','color',colors(sp_k,:)); hold on;

end

set(gcf, 'Position',  [180, 240, 1280, 360])

annotation('textbox',[0.65 0.7 .15 .25],'BackgroundColor','w');
annotation('textbox',[0.65 0.65 .3 .3],'String','\bullet Minimum Energy Principle',
            'FitBoxToText','on','EdgeColor','none','color','b');
annotation('textbox',[0.65 0.6 .3 .3],'String','\bullet Armarego Model','FitBoxToText','on',
            'EdgeColor','none','color','r');
annotation('textbox',[0.65 0.55 .3 .3],'String','\bullet Simplified Armarego Model','FitBoxToText',
           'on','EdgeColor','none','color','m');
annotation('textbox',[0.65 0.5 .3 .3],'String','* Lin & Oxley (1972)','FitBoxToText','on',
            'EdgeColor','none','color','k');
annotation('textbox',[0.825 0.8 .15 .15],'BackgroundColor','w');
annotation('textbox',[0.825 0.65 .3 .3],'String','* U = 60.96 m/min','FitBoxToText','on',
            'EdgeColor','none','color',colors(2,:));
annotation('textbox',[0.825 0.6 .3 .3],'String','* U = 121.92 m/min','FitBoxToText','on',
            'EdgeColor','none','color',colors(3,:));
```

And the Final plot we get is as follows:

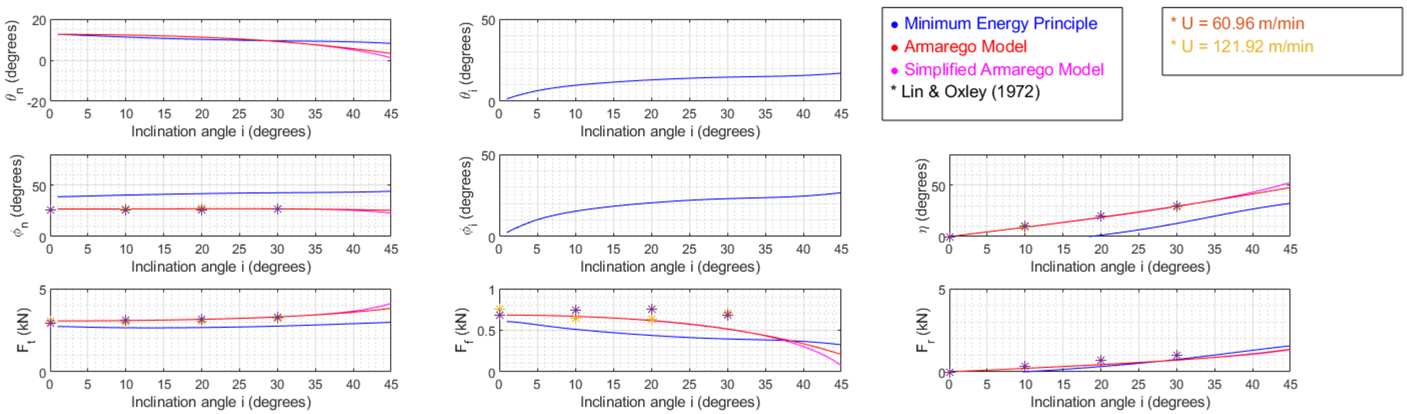

Figure 1: Final plot of question 1

Thus, the oblique cutting parameters predicted by the proposed solutions are in good agreement with the empirical and experimental data published in the literature.

References

1. Altintas, Y. (2012). Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design (2nd ed.). Cambridge: Cambridge University Press. doi:10.1017/CBO9780511843723

2. Lin GCI, Oxley PLB. Mechanics of Oblique Machining: Predicting Chip Geometry and Cutting Forces from Work Material Properties and Cutting Conditions. Proceedings of the Institution of Mechanical Engineers. 1972;186(1):813-820. doi:10.1177/002034837218600151

3. Shamoto, E., and Altıntas, Y. (August 1, 1999). "Prediction of Shear Angle in Oblique Cutting with Maximum Shear Stress and Minimum Energy Principles." ASME. J. Manuf. Sci. Eng. August 1999; 121(3): 399–407. https://doi.org/10.1115/1.2832695

4. Law, M., Class Lecture, Topic: "Oblique Cutting." ME668A, Indian Institute of Technology, Kanpur, Jan., 2021.