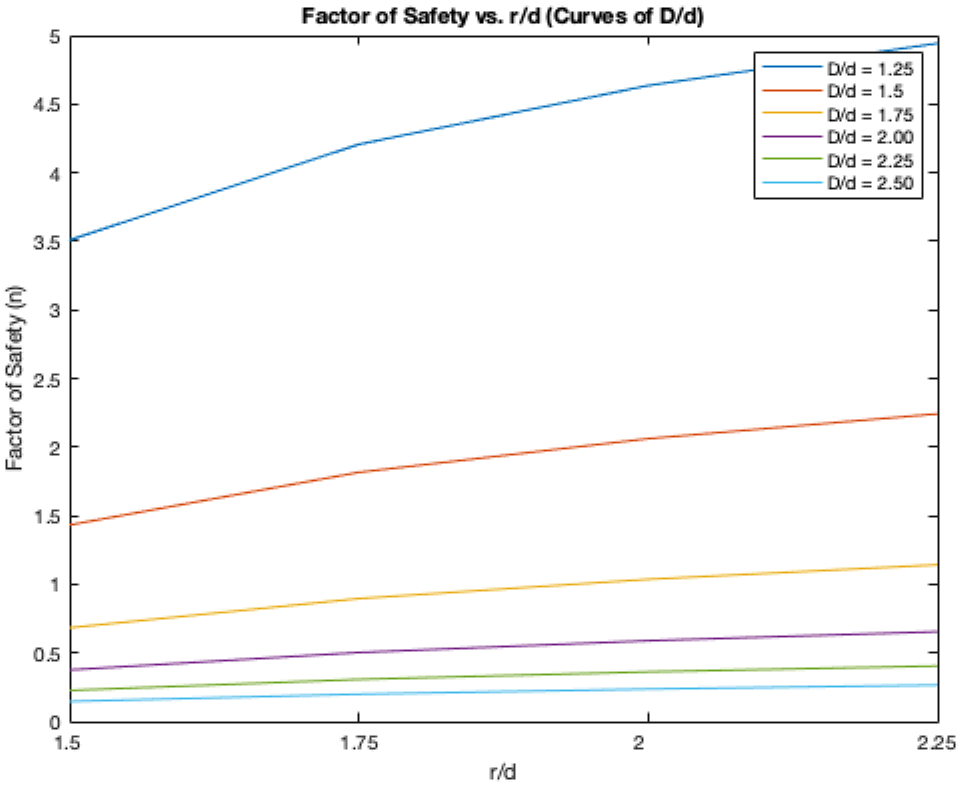
```
% NASH ELDER %
% ME 328 DESIGN I, DR. DAV0L %

% FATIGUE %
% Equations and recommended values from Shigley's 9th ed. %

% MATERIAL = STEEL %

clear;

Sy = 60.2; % ksi %
Sut = 95; % ksi %

Pa = 0.2; % kip %
Pm = 0.8; % kip %
P = 1;
Dbig = 3.5; % in. %
L = 15; % in. %
a = 8; % in. %

aSurf = 2.70; % Finish Factor a (machined finish) %
bSurf = -0.265; % Finish Factor b (machined finish) %

SePrime = 0.5*Sut; % ksi -- For Sut <= 200 %

for i = 1:6
    Doverd = 1.25 + 0.25*(i-1);
    d = (Dbig/Doverd);

    for j = 1:4
        roverd = 0.05 + 0.05*(j-1);
        r = (roverd * d);

        Ka = aSurf*Sut^bSurf;

        if d <= 2.0

            Kb = 0.879*(0.370*d)^(-0.107); % Size Factor -- small
 radius %
        else
            Kb = 0.91*(0.370*d)^(-0.157); % Size Factor -- large
 radius %
        end

        Kc = 1; % Load Factor -- Bending Load %

        Kd = 1; % Temp Factor -- Temp = 20 deg. C %

        Za = 1.288; % Transformation Variate for 90% (TBL 6-5) %

        Ke = 1 - 0.08*Za; % Reliability Factor %
        Se = SePrime * Ka * Kb * Kc * Kd * Ke;

        % Kt due to bending and torsion (ref: AMESWEB.INFO) %

        h = (Dbig-d)/2;

        if 0.1 <= h/r <= 2.0
            c1b = 0.947+1.206*(sqrt(h/r))-0.131*h/r;
            c2b = 0.022-0.3405*(sqrt(h/r))+0.915*h/r;
            c3b = 0.869+1.777*(sqrt(h/r))-0.555*h/r;
            c4b = -0.810+0.422*(sqrt(h/r))-0.260*h/r;
        elseif 2.0 < h/r <= 20.0
            c1b = 1.232+0.832*(sqrt(h/r))-0.008*h/r;
            c2b = -3.813+0.968*(sqrt(h/r))-0.260*h/r;
            c3b = 7.423-4.868*(sqrt(h/r))+0.869*h/r;
            c4b = -3.839+3.070*(sqrt(h/r))-0.600*h/r;
        else
            c1b = NaN; c2b = NaN; c3b = NaN; c4b = NaN;
        end

        Kt = c1b + c2b*(2*h/Dbig) + c3b*(2*h/Dbig)^2 + c4b*(2*h/
Dbig)^3;

        if 0.25 <= h/r <= 4.0
            c1t = 0.905+0.783*(sqrt(h/r))-0.075*h/r;
            c2t = -0.437-1.969*(sqrt(h/r))+0.553*h/r;
            c3t = 1.557+1.073*(sqrt(h/r))-0.578*h/r;
            c4t = -1.061+0.171*(sqrt(h/r))+0.086*h/r;
        else
            c1t = NaN; c2t = NaN; c3t = NaN; c4t = NaN;
        end

        Kts = c1t + c2t*(2*h/Dbig) + c3t*(2*h/Dbig)^2 + c4t*(2*h/
Dbig)^3;

        % q calculation to find Kf %

        sqrtA = 0.246-3.08*10^(-3)*Sut
+1.51*10^(-5)*Sut^2-2.67*10^(-8)*Sut^3;
        sqrtA_tor = 0.190-2.51*10^(-3)*Sut
+1.35*10^(-5)*Sut^2-2.678*10^(-8)*Sut^3;


        q = 1/(1+sqrtA/sqrt(r));
        qs = 1/(1+sqrtA_tor/sqrt(r));

        Kf = q * (Kt-1) + 1;
        Kfs = qs * (Kts-1) + 1;

        % Sigma calculation %

        Ma = Pa * L;
        Mm = Pm * L;
        Ta = Pa * a;
        Tm = Pm * a;
        I = pi/4*(d/2)^4;
        J = pi/2*(d/2)^4;

        sig_a_bend = Ma*(d/2)/I;
        sig_m_bend = Mm*(d/2)/I;
        tau_a = Ta*(d/2)/J;
        tau_m = Tm*(d/2)/J;

        sig_prime_a = ((Kf*sig_a_bend)^2 + 3*(Kfs*tau_a)^2)^(0.5);
        sig_prime_m = ((Kf*sig_m_bend)^2 + 3*(Kfs*tau_m)^2)^(0.5);

        sig_prime_max = sig_m_bend + sig_a_bend;
        tau_prime_max = tau_a + tau_m;

        % Safety factors %

        n_fatigue = 1/((sig_prime_a/Se) + (sig_prime_m/Sut));% Safety
 factor for fatigue %

        n_SF = Sy / sig_prime_max; % Safety factor Yield / Allowable %

        n = min(n_fatigue, n_SF);

        SF(j,i) = n; % Allocation of safety factors on each iteration
 (for plot) %

    end
end
disp('4x6 Vector of 6 different D/d values and 4 different r/d
 values:');
disp(SF);

plot(SF);
xlabel('r/d');
ylabel('Factor of Safety (n)');
legend('D/d = 1.25', 'D/d = 1.5', 'D/d = 1.75', 'D/d = 2.00', 'D/d =
 2.25', 'D/d = 2.50');
set(gca, 'XTick', 0:1:4);
set(gca, 'XTickLabel', 1.25:0.25:2.5);
title('Factor of Safety vs. r/d (Curves of D/d)');
```

```
4x6 Vector of 6 different D/d values and 4 different r/d values:
    3.5085    1.4323    0.6830    0.3759    0.2266    0.1462
    4.2066    1.8161    0.8957    0.5029    0.3067    0.1993
    4.6352    2.0614    1.0363    0.5888    0.3619    0.2364
    4.9427    2.2422    1.1420    0.6545    0.4047    0.2653
```



*Published with MATLAB® R2018b*