Universitat Politècnica de Catalunya
Numerical Methods in Engineering
Computational Solid Mechanics

1D computational plasticity

Eduard Gómez
April 22, 2020

Contents

1 Inviscid plasticity **1**
1.1 Perfect plasticity . . . 1
1.2 Linear plasticity . . . 2
1.3 Non-linear isotropic plasticity . . . 4

2 Viscous plasticity **6**
2.1 Effect of loading rate . . . 6
2.2 Effect of viscous parameter . . . 8
2.3 All models combined . . . 9

A Appendix **10**
A.1 Main file . . . 10
A.2 Hardening . . . 12
A.3 Pi function . . . 12
A.4 Post-processing . . . 13
A.5 Strain-stress plot . . . 15

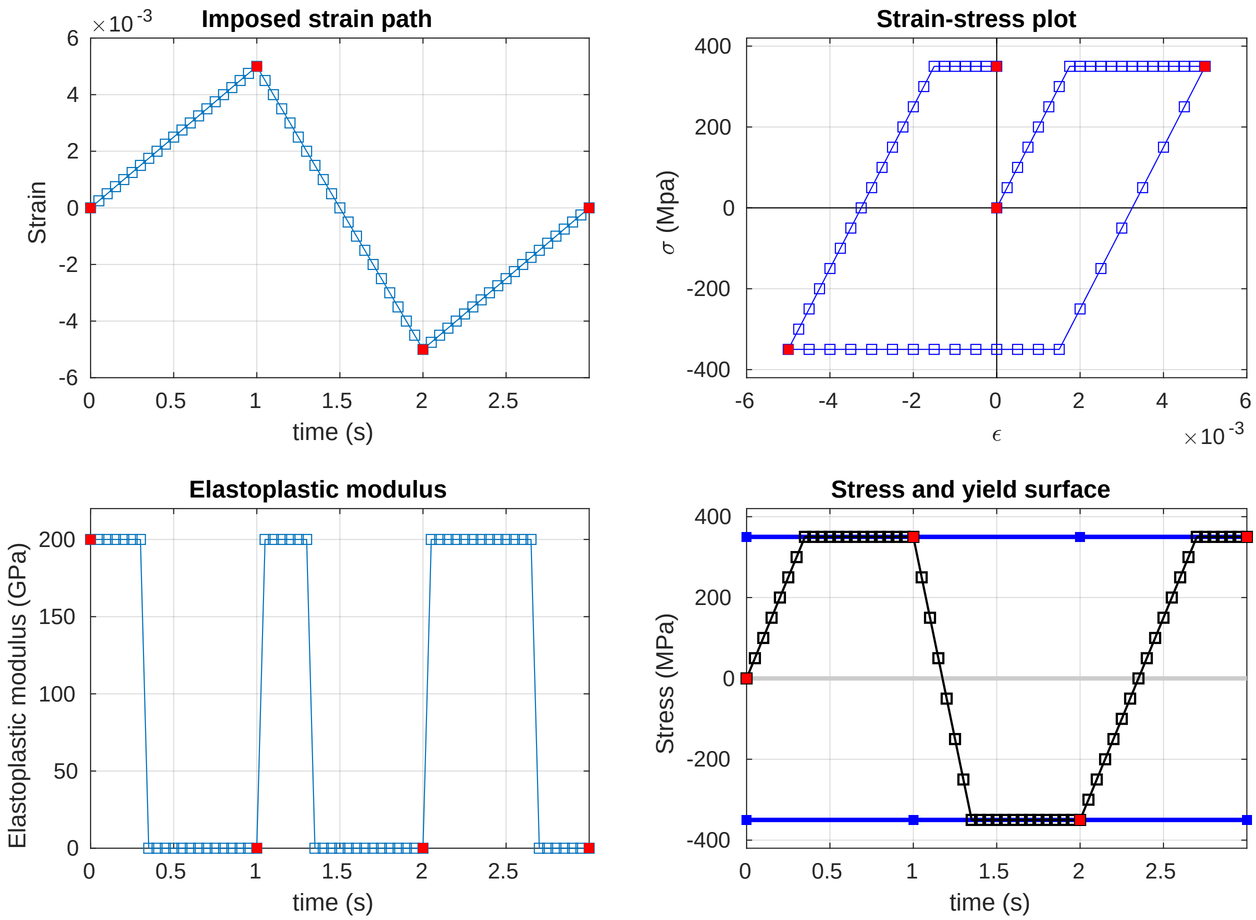
1 Inviscid plasticity

1.1 Perfect plasticity

Perfect plasticity is characterized by having an unchanging yield surface that limit the space of possible stresses. During plastic deformation stress remains constant, meaning that the elastoplastic modulus is zero. All these phenomena can be seen in figure 1.

Note that although the Elastic modulus and yield stress mimic those of steel, all other properties are chosen to magnify their respective effects in the figures, all the while respecting thermodynamic constraints.

All figures follow this format, so it is worth stopping and explaining them. The top left quadrant has the imposed strain path with some red squares that act as reference points. These can be seen in all other figures to facilitate their comparison. The second and third sub figures are quite self-explanatory. The last plot has the stress evolution (black line with squares) and the yield surface (two blue lines: traction and compression).



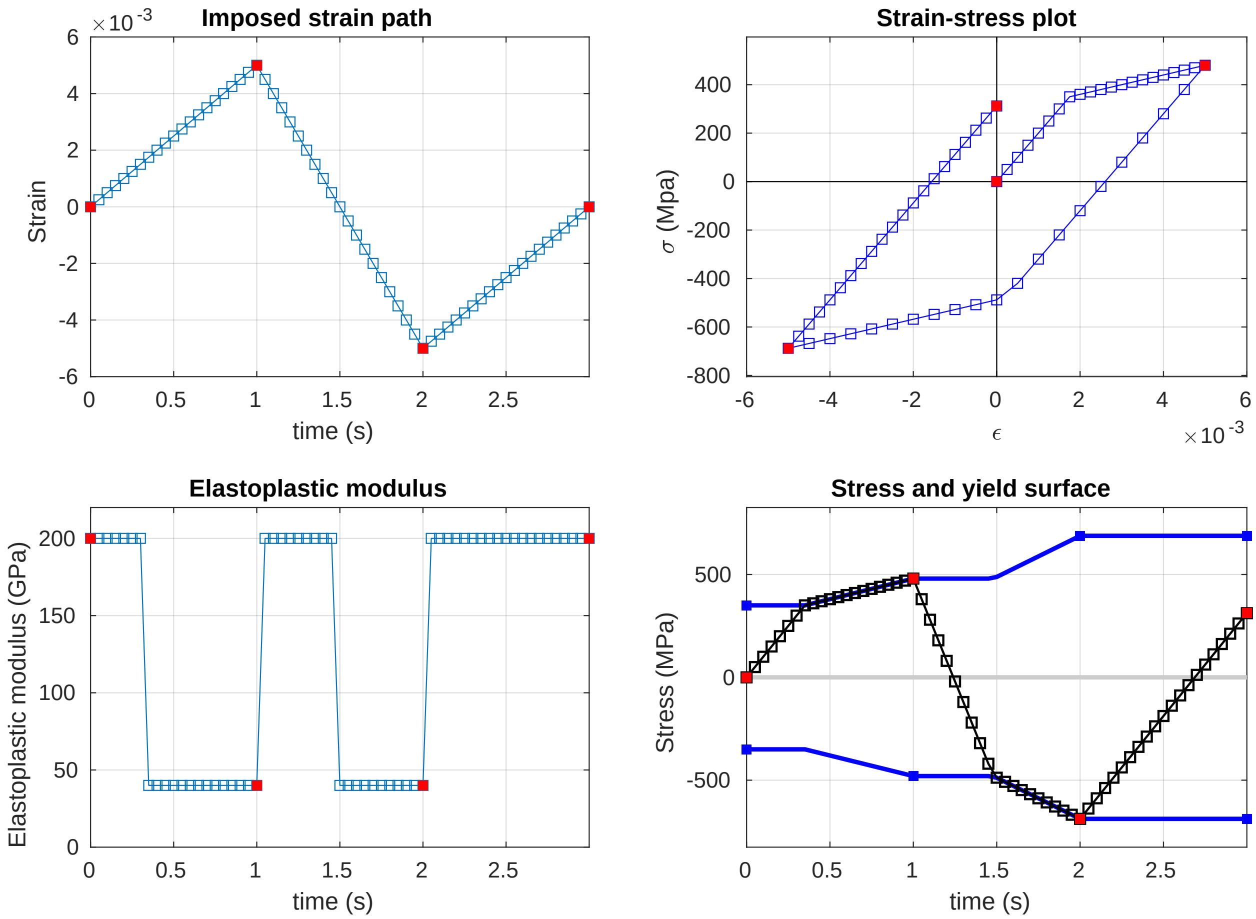
$E = 200$ GPa $\qquad \sigma_y = 350$ Mpa $\qquad H = K = \eta = 0$

Figure 1: Perfect plasticity analysis

1.2 Linear plasticity

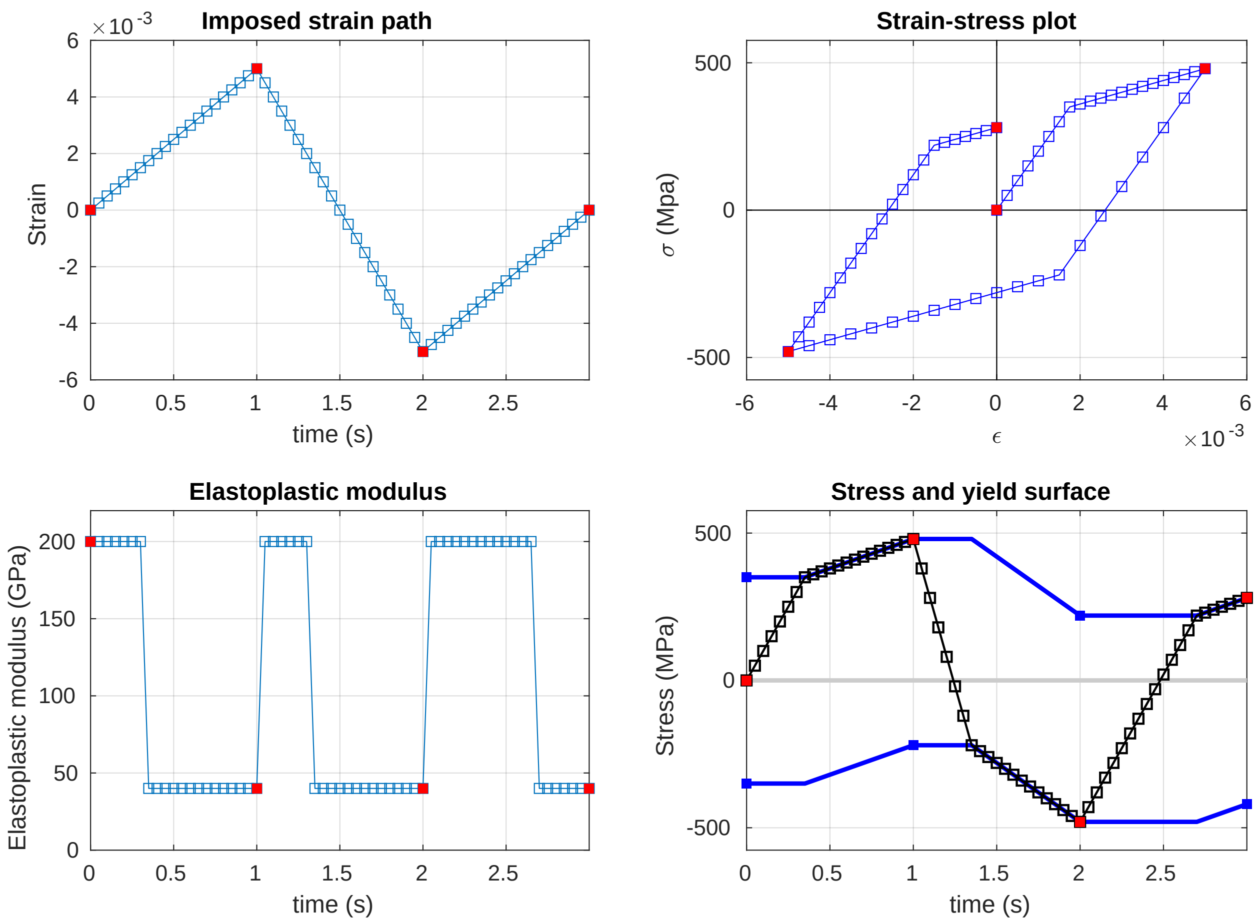
The first change we can apply is considering linear plasticity. The implication is that the yield surface can change. This phenomenon is called hardening. We can consider isotropic hardening, where the surface expands symmetricaly; and kinematic hardening, where the surface is displaced but its width is preserved. The former can be seen in figure 2, and the latter in figure 3. In both cases, the elastoplastic modullus is expected to decrease from its linear counterpart, but stay above zero unlike perfect plasticity.

It is worth noting that, although similar, the values of final stress for perfect, linear isotropic hardening, and linear kinematic hardening plasticity are all different.



$E = 200$ GPa $\sigma_y = 350$ Mpa $K = 50$ GPa $H = \eta = 0$

Figure 2: Linear isotropic plasticity analysis



$E = 200$ GPa $\sigma_y = 350$ Mpa $K = 50$ GPa $H = \eta = 0$

Figure 3: Linear kinematic plasticity analysis

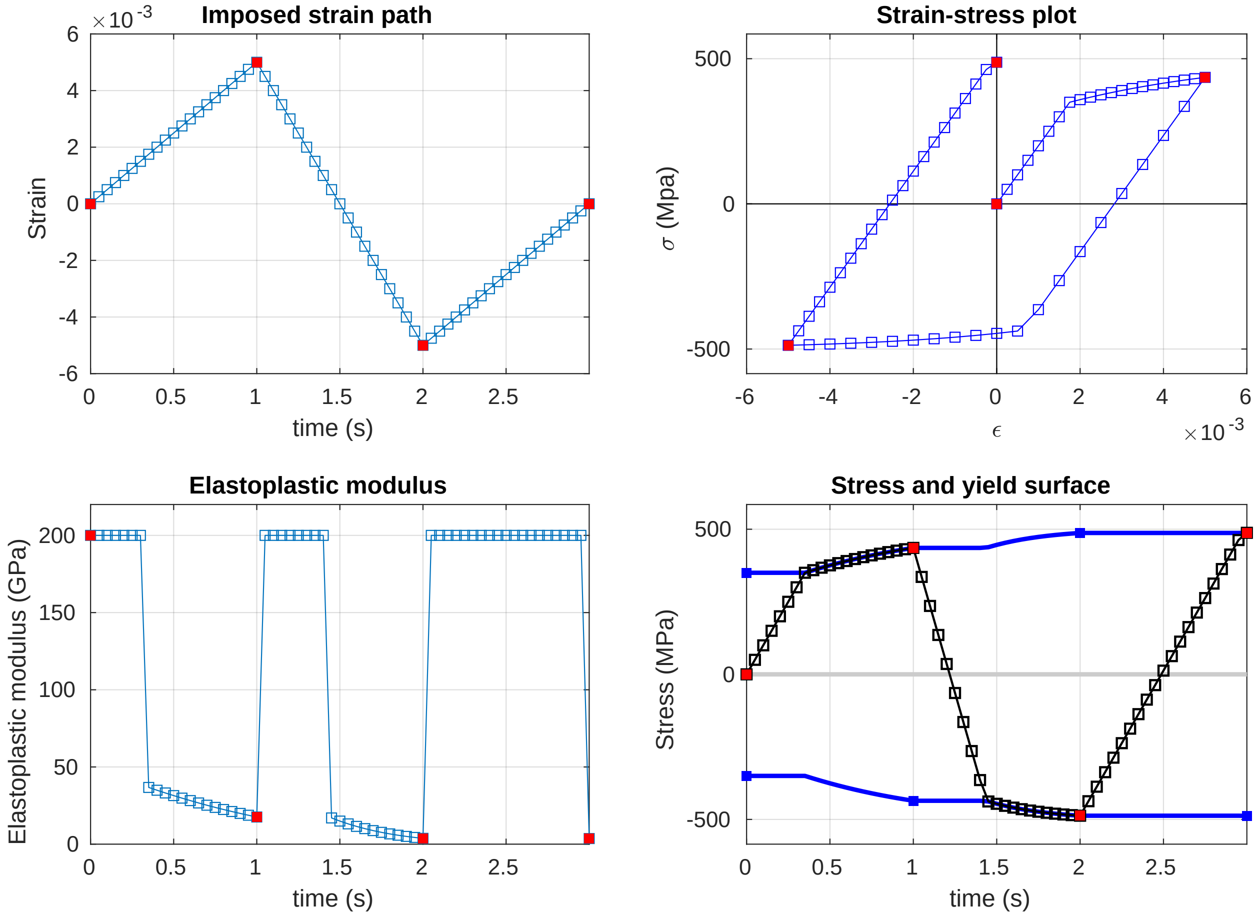
1.3 Non-linear isotropic plasticity

A further addition of complexity to the model is the inclusion of a non linear model for isotropic hardening. In our case we consider:

$$q = (\sigma_\infty - \sigma_y)(1 - \exp(-\delta\xi)) + K\xi \tag{1}$$

where q is the symmetrical increase of the yield surface and ξ is its counterpart in strain space. This model introduces two new paramenters δ and σ_∞ to model the exponential curve. The first one controls the rate of increase whereas the second one locates the exponential assymptote. To make these effect more visible we can set $K = 0$.

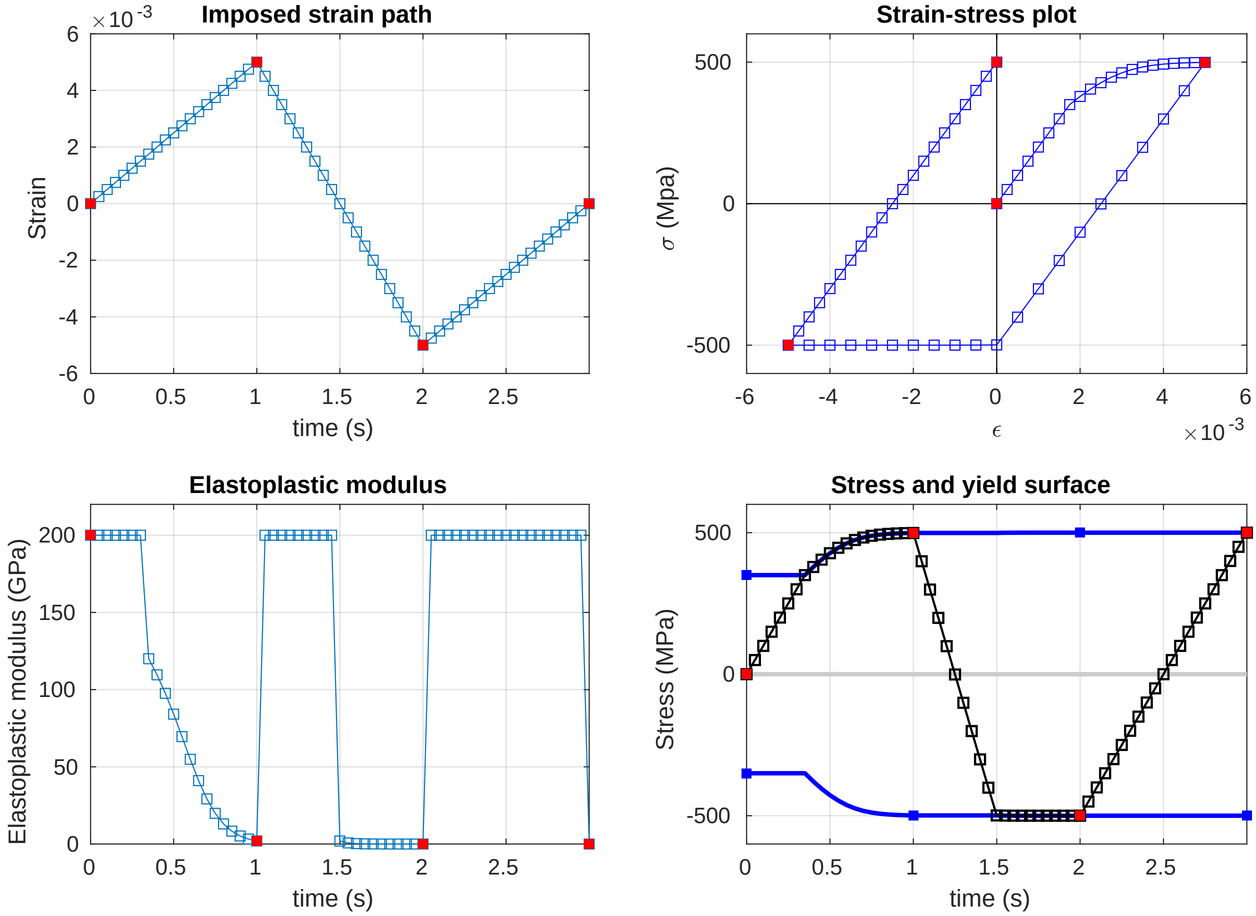
Figures 4 and 5 offer a comparison for two different values of δ. In the first case, with $\delta = 300$, the stress required to increase the strain (i.e. the elastoplastic modulus) is quite low, so the assimptotic value of $\sigma_\infty = 500$ MPa is only reached at the end of the compressive stage. The result is something simmular to linear model, but where the elastoplastic modulus decreases the larger the plastic deformation.



$E = 200$ GPa $\quad \sigma_y = 350$ Mpa $\quad \delta = 300 \quad \sigma_\infty = 500$ MPa $\quad K = H = \eta = 0$

Figure 4: Non-linear isotropic plasticity analysis with small δ

In the second case, with $\delta = 2 \times 10^3$, stress grows much faster, so the assymptotic values is quickly reached (for all intents and purposes) during the first traction stage. After that, the model behaves the same as perfect plasticity. Had K or H been different than zero, the model would have started behaving as linear plasticity. In summary, when $\sigma \to \sigma_\infty$, the linear (or perfect) underlying model dominates.



$E = 200$ GPa $\quad \sigma_y = 350$ Mpa $\quad \delta = 2 \times 10^3 \quad \sigma_\infty = 500$ MPa $\quad K = H = \eta = 0$

Figure 5: Non-linear isotropic plasticity analysis with large δ

2 Viscous plasticity

A viscous model includes a new parameter η. This parameters acts as a coefficient that opposes the change of ε^p over time. Before now, the stresses all went towards elastic and plastic strains, but now part of the stress will go towards fighting viscosity (only when in the plastic domain). In a simplified way this can be expressed as:

Non viscous model	Viscous model
$\sigma = \sigma^e(\varepsilon^e) + \sigma^p(\varepsilon^p)$	$\sigma = \sigma^e(\varepsilon^e) + \sigma^p(\varepsilon^p) + \sigma^\eta(\dot{\varepsilon}^p)$

The sum of the first two terms is bounded by the yield surface. The addition of the viscous term means that the space of admissible stresses will grow beyond the yield surface.

2.1 Effect of loading rate

When outside the yield surface, and given a fixed strain and enough time, the yield surface will grow and the viscous component will decrease until the stress is once again inside the yield surface (see figure 6). The time this will take depends on the viscosity. We can see how stress spikes up to 780 MPa.

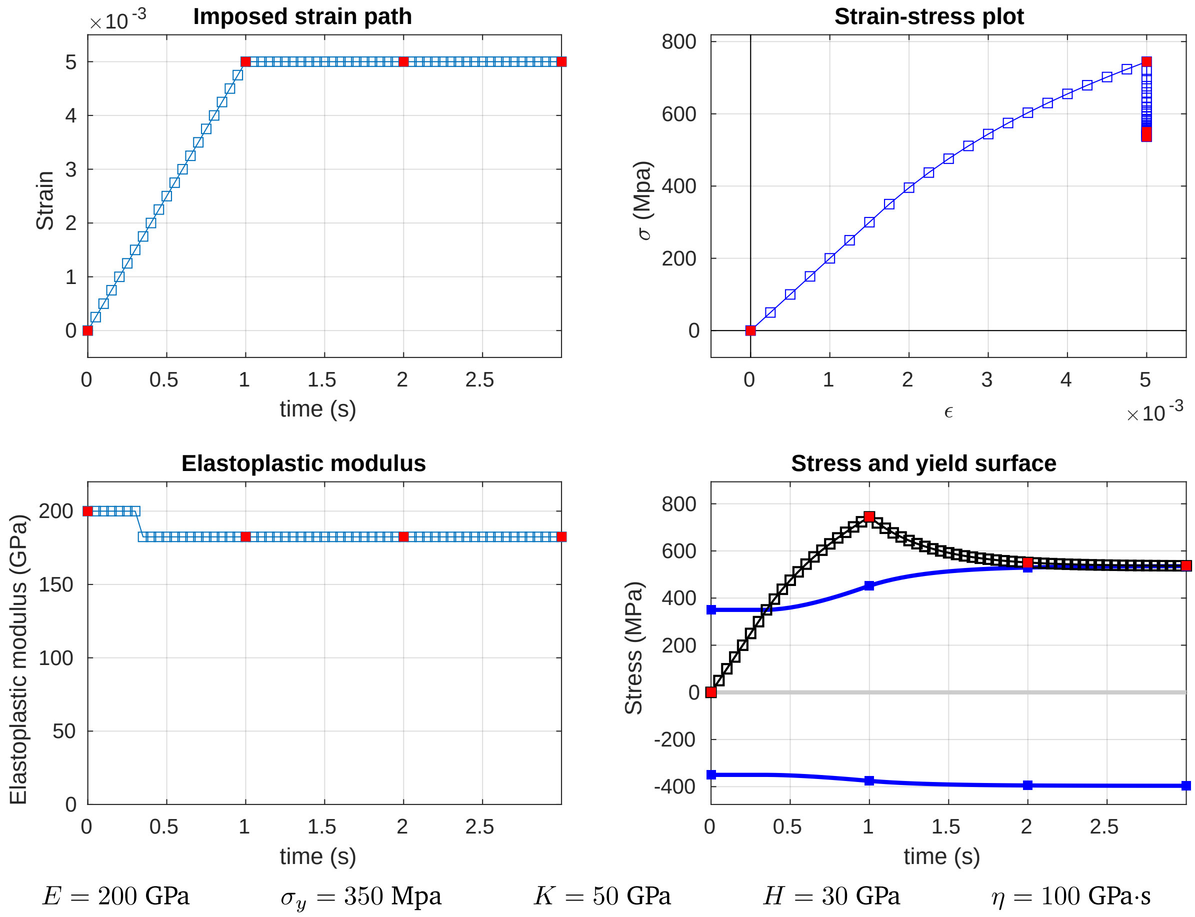
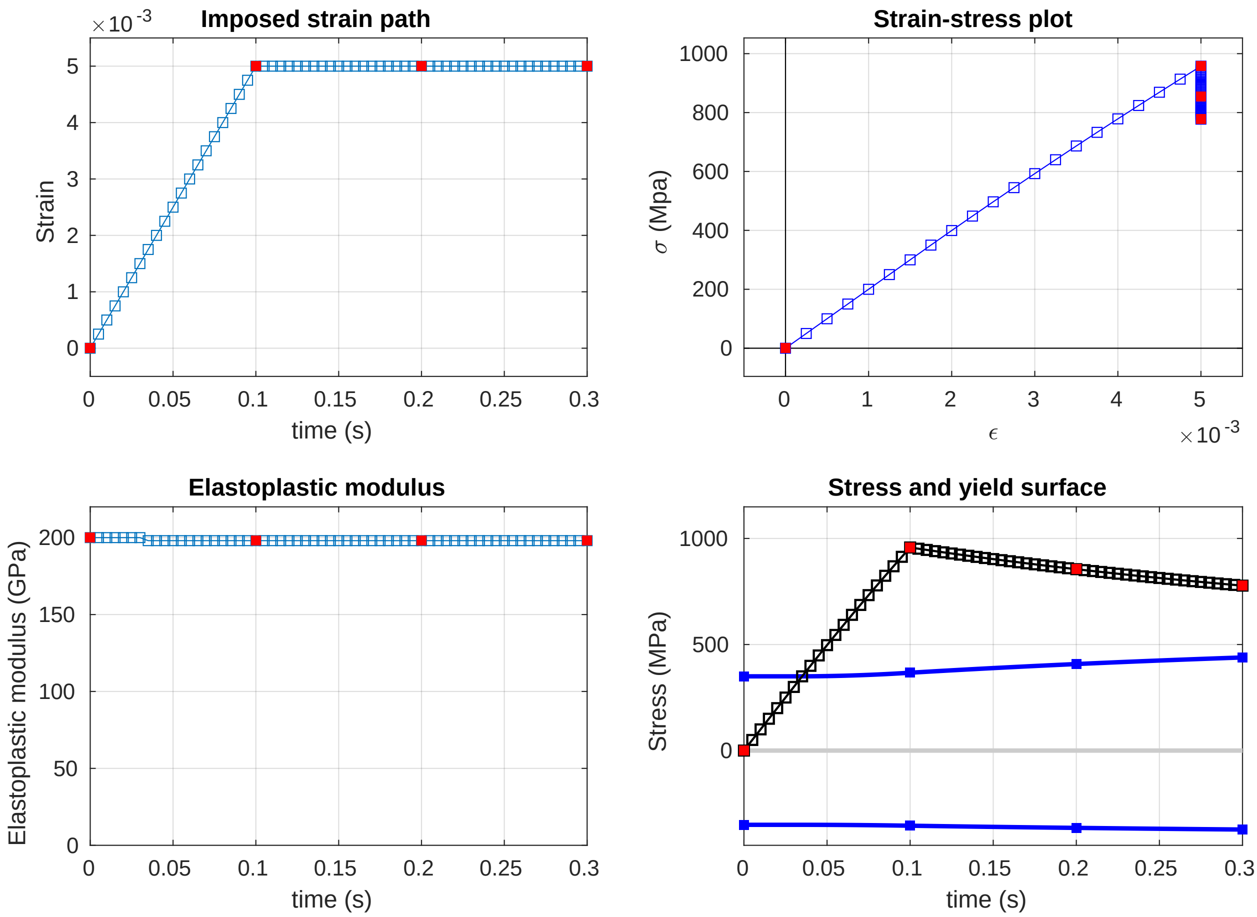


Figure 6: Linear hardening viscous plasticity analysis

Figure 7 shows the same process performed ten times faster. We can observe how the stress spike is 25% higher.

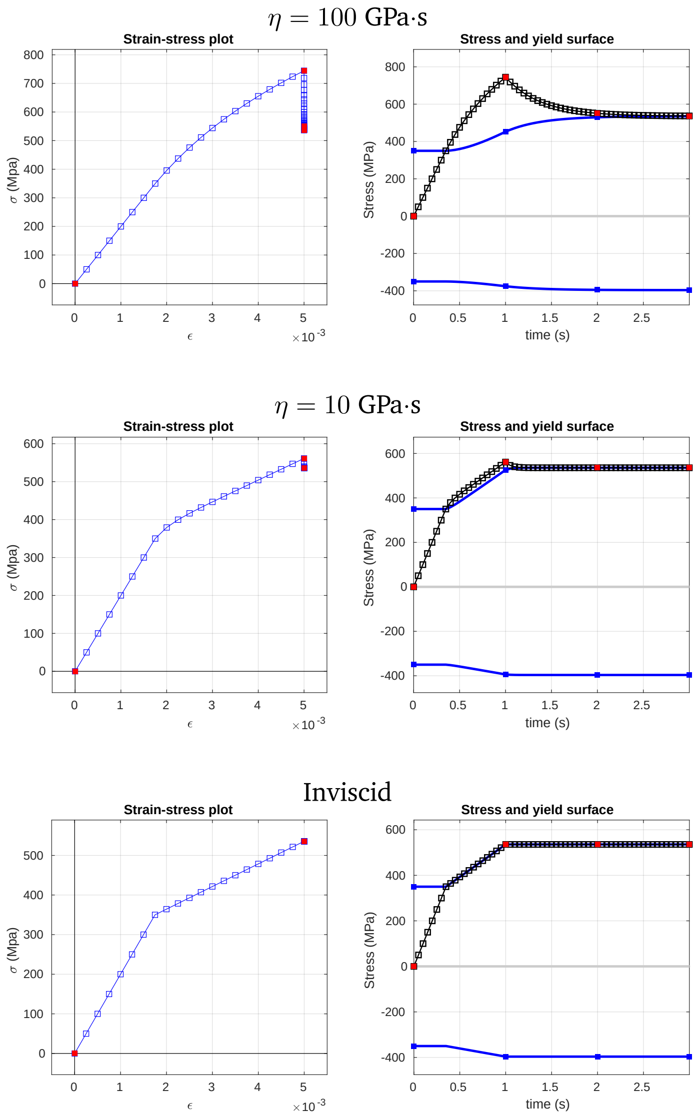


$E = 200$ GPa $\sigma_y = 350$ Mpa $K = 50$ GPa $H = 30$ GPa $\eta = 100$ GPa·s

Figure 7: Linear hardening viscous plasticity analysis

2.2 Effect of viscous parameter

If we repeat the same process in with different viscosities we see that as viscosity decreases the system approaches an inviscid one. This can be seen in figure 8.



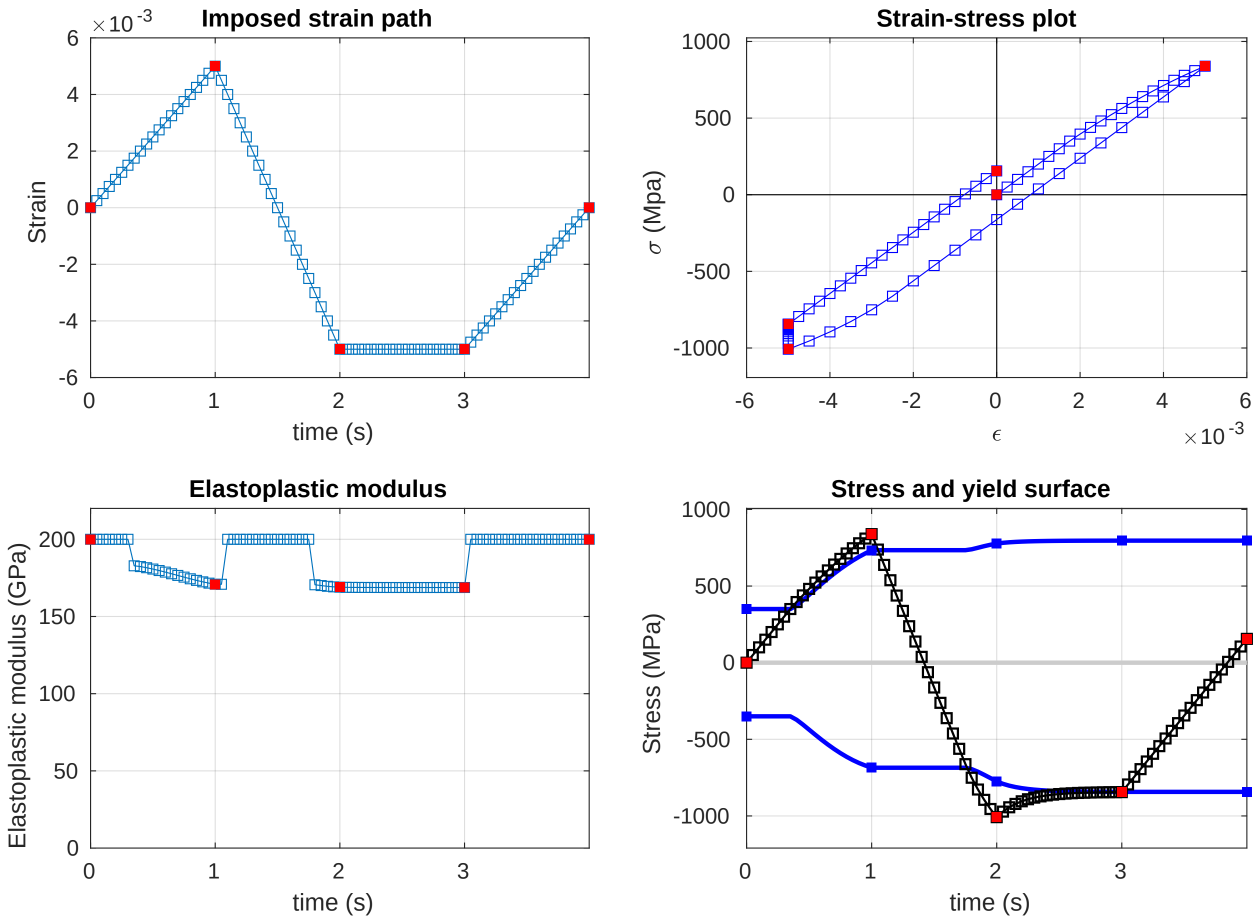
$E = 200$ GPa $\sigma_y = 350$ Mpa $K = 50$ GPa $H = 30$ GPa

Figure 8: Effect of diminishing viscosity

2.3 All models combined

To finish it up, here is an example of viscous, non-linear isotropic, linear kinematic viscosity. The effects can be identified by:

- Viscosity: Stress path is outside the yield surface, and slowly returns to it when strain remains constant.
- Exponential component of isotropic hardening: The elastoplastic modulus approaches an asymptote when in the plastic domain.
- Linear component of isotropic hardening: the aforementioned asymptote is greater than zero.
- Kinematic hardening: The yield surface is slightly off-centre (-873 MPa vs. 818 MPa at $t = 4$ s).



$E = 200$ GPa $\quad \sigma_y = 350$ Mpa $\quad H = 30$ GPa $\quad \eta = 50$ GPa·s

$K = 50$ GPa $\quad \delta = 3 \times 10^3 \quad \sigma_\infty = 700$ MPa

Figure 9: Viscous, non-linear isotropic, linear kinematic viscosity.

A Appendix

A.1 Main file

```
%% Data entry
% Material properties
mat.E = 200e3;
mat.yield = 350;
mat.K = 50e3;
mat.H = 30e3;
mat.visc = 50e3;

% Strain path
% Defined by an arbitrary number of 'trips'.
% Each trip takes the strain from one corner value to the next one.
% The corners are marked in red in the plots.
corners = [0, 0.005, -0.005, 0];
steps_per_trip = 15;
time_per_trip = 1;

% Hardening
hardening.is_linear = false;

mat.sigma_infty = 700;
mat.delta = 3000;

hardening.maxIter = 30;
hardening.tol = 1e-10;

%% Pre-processing
% Support variables
n_steps = steps_per_trip * (length(corners)-1) + 1;
dt = time_per_trip / steps_per_trip;
mat.C = diag([mat.E, mat.K, mat.H]);

% Computing full strain path
strain = zeros(1,n_steps);
for i = 1:length(corners)-1
    start = (i-1)*steps_per_trip + 1;
    ending = i*steps_per_trip;
    trip = linspace(corners(i), corners(i+1), steps_per_trip+1);
    strain(start:ending) = trip(1:end-1);
end
strain(end) = corners(end);

% Initialization of arrays
Strain_p = [zeros(1,n_steps)   % Strain
            zeros(1,n_steps)   % Xi
```

```
            zeros(1,n_steps)]; % Xi bar

Stress = [zeros(1,n_steps)   % sigma
          zeros(1,n_steps)   % q
          zeros(1,n_steps)]; % q bar

elastoplastic_modulus = zeros(n_steps,1);
elastoplastic_modulus(1) = mat.E;

%% Computation
for i = 2:n_steps
    Strain_next = [strain(i);0;0];
    S_trial = mat.C * (Strain_next - Strain_p(:,i-1));

    if hardening.is_linear == false
         S_trial(2) = Stress(2,i-1);
    end

    f_trial = abs(S_trial(1) - S_trial(3)) - mat.yield + S_trial(2);

    if f_trial < 0
        % Elastic load/unload
        Strain_p(:,i) = Strain_p(:,i-1);
        Stress(:,i) = S_trial;
        elastoplastic_modulus(i) = mat.E;
    else
        % Plastic load
        f_grad(1,1) = sign(S_trial(1) - S_trial(3));
        f_grad(2,1) = 1;
        f_grad(3,1) = - f_grad(1,1);

        [gamma, EPM] = calc_hardening(hardening, Strain_p(:,i-1), f_trial, mat, dt);

        Stress(:,i) = S_trial - gamma*mat.C*f_grad;
        Strain_p(:,i) = Strain_p(:,i-1) + gamma*f_grad;

        if hardening.is_linear==false
            xi = Strain_p(2,i);
            Stress(2,i) = - Pi(xi,mat,1);
        end

        elastoplastic_modulus(i) = EPM;
    end
end

run post_processing
```

A.2 Hardening

```
function [gamma,EPM] = calc_hardening(hardening, Strain_p, f_trial, mat, dt)

    if hardening.is_linear
        gamma = f_trial / (trace(mat.C) + mat.visc/dt);
        EPM = mat.E * (1 - mat.E / (trace(mat.C) + mat.visc/dt));
    else
        gamma = 0;
        xi = Strain_p(2);
        for k=1:hardening.maxIter
            g = f_trial - gamma * (mat.E + mat.H + mat.visc/dt) ...
                - (Pi(xi + gamma, mat, 1) - Pi(xi,mat,1));
            if(abs(g) < hardening.tol)
                break
            end
            Dg = -(mat.E + mat.H + mat.visc/dt + Pi(xi + gamma,mat,2));
            gamma = gamma - g/Dg;
        end
        if(k==hardening.maxIter)
           warning(['Maximum number of iterations reached' ...
                     'before convergence. Error %f'],abs(g))
        end
        % Elastoplastic Modulus
        EPM = mat.E * (1 - mat.E / (mat.E + Pi(xi+gamma,mat,2) + mat.H + mat.visc/dt));
    end
end
```

A.3 Pi function

```
function z = Pi(xi, mat, derivative)
    switch derivative
        case 1
            % Pi'(xi)
            z = (mat.sigma_infty - mat.yield)*(1-exp(-mat.delta*xi)) + mat.K*xi;
        case 2
            % Pi"(xi)
            z = mat.delta*(mat.sigma_infty - mat.yield)*exp(-mat.delta*xi) + mat.K;
        otherwise
            error('Only first and second derivatives implemented')
    end
end
```

A.4 Post-processing

```
t = dt * ones(n_steps,1);
t = cumsum(t) - dt;
corner_indices = 1:steps_per_trip:n_steps;

figure('Renderer', 'painters', 'Position', [10 10 900 600])

%% Strain evolution
subplot(221);
plot(t, strain,'s-');
hold on
scatter(t(corner_indices), corners,'r','square', 'filled')
hold off

grid on
width = 0.1*(max(strain) - min(strain));
axis([0, t(end), min(strain)-width, max(strain)+width]);
xlabel('time (s)');
ylabel('Strain');
title('Imposed strain path');

%% Strain-stress
subplot(222);
plot_strain_stress(strain, Stress(1,:));
hold on
scatter(strain(corner_indices), Stress(1,corner_indices),'r','square', 'filled')
hold off

%% Elastoplastic modulus
subplot(223);
ep = elastoplastic_modulus/1e3;

plot(t, ep,'s-');
hold on
scatter(t(corner_indices), ep(corner_indices),'r','square', 'filled')
hold off

grid on
axis([0 t(end) 0 max(ep)*1.1]);

xlabel('time (s)');
ylabel('Elastoplastic modulus (GPa)');
title('Elastoplastic modulus');

%% Stress and yield evolution
subplot(224);
yield_traction = mat.yield - Stress(2,:) + Stress(3,:);
```

```
yield_compression = - (mat.yield - Stress(2,:) - Stress(3,:));

% Plotting evolution
plot(t, yield_compression,'b-','LineWidth',2);
hold on
plot(t, yield_traction,'b-','LineWidth',2,'HandleVisibility','off');
plot([-1, t(end)*1.1],[0 0],'Color', [1 1 1]*0.8,'LineWidth',2,'HandleVisibility','off');
plot(t,Stress(1,:),'ks-','LineWidth',1);

% Plotting corner points
scatter(t(corner_indices), yield_traction(corner_indices),'b','square', 'filled')
scatter(t(corner_indices), yield_compression(corner_indices),'b','square', 'filled')
scatter(t(corner_indices), Stress(1,corner_indices),'r','square', 'filled')
hold off

% Labels and appearance
grid on
min_y = min([yield_compression, Stress(1,:)])*1.2;
max_y = max([yield_traction, Stress(1,:)])*1.2;
axis([t(1), t(end)  min_y max_y]);
xlabel('time (s)');
ylabel('Stress (MPa)');
title('Stress and yield surface');

%% Saving figure
saveas(gca, 'figure.pdf');
system('pdfcrop figure.pdf figure.pdf');
system('mv figure.pdf Figures/figure.pdf');
```

A.5 Strain-stress plot

```
function plot_strain_stress(strain,stress)
    % Calculating x-range
    x_width = max(strain) - min(strain);
    x_lim(1) = min(strain) - 0.1 * x_width ;
    x_lim(2) = max(strain) + 0.1 * x_width ;

    % Calculating y-range
    y_width = max(stress) - min(stress);
    y_lim(1) = min(stress) - 0.1 * y_width;
    y_lim(2) = max(stress) + 0.1 * y_width;

    % Plotting axes
    plot(x_lim,[0 0],'k');
    hold on
    plot([0 0],y_lim,'k');

    % Plotting stress-strain
    plot(strain, stress,'sb-');
    hold off

    % Labels, grid, etc
    grid on
    axis([x_lim, y_lim]);
    xlabel('\epsilon');
    ylabel('\sigma (Mpa)');
    title('Strain-stress plot');
end
```