TP de Plasticité
Modèle 1D pour la plasticité
Rapport

Muruo Wang

Septembre 2018

1 Préliminaire : algorithme de retour radical

L'algorithme de retour radical illustre la méthode pour le modèle élasto-plastique parfait.
$\epsilon^{n+1} = \epsilon^n + d\epsilon$
Prédiction élastique :
$\sigma_{pred} = E(\epsilon^{n+1} - \epsilon_p^n)$
$f_{pred} = |\sigma_{pred}| - \sigma_y$

```
for i=1:N-1

epsilon(i+1)=epsilon(i)+d_epsilon(i) ;
sigma_pred=E*(epsilon(i+1)-epsilon_p(i));
f_pred= abs(sigma_pred)-sigma_y;

 if (f_pred < 0) %élastique
    lambda=0;
    epsilon_p(i+1)=epsilon_p(i);
    sigma(i+1)=sigma_pred;
 else              %plasticité
     lambda=f_pred/E;
     epsilon_p(i+1)=epsilon_p(i)+sign(sigma_pred)*lambda;
     sigma(i+1)=E*(epsilon(i+1)-epsilon_p(i+1));
 end
end
```

— ϵ_n au point(1), $d\epsilon > 0$ et petit(σ_y n'est pas franchie)
Ici, j'ai fixé d$\epsilon = 0.01$, $f_{pred} = 0.01$, $\sigma_{pred} = 2.01$
— ϵ_n au point(1), $d\epsilon < 0$ et petit
Ici, j'ai fixé d$\epsilon = -0.01$, $f_{pred} = 0.01$, $\sigma_{pred} = -2.01$
— ϵ_n au point(2), $d\epsilon > 0$ et σ_y est franchie
— ϵ_n au point(3), $d\epsilon > 0$
— ϵ_n au point(3), $d\epsilon < 0$

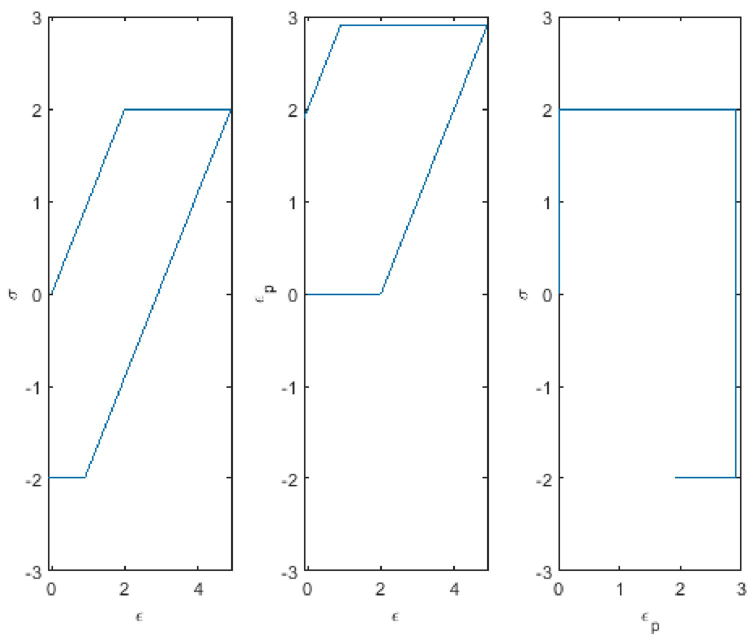
2 Etude du modèle élasto-plastique parfait

On choisit tout d'abord des valuers simples : $E = 1$ MPa et $\sigma_y = 2$MPa

2.1 Programmer l'algorithme et le tracé des courbes

```
% 2.a.
subplot(131);
plot(epsilon,sigma)   ;

% 2.b. charge-décharge en traction  epsilon=0   70ème=0(y=0)
%plot(epsilon(1:70),sigma(1:70));

xlabel("\epsilon");ylabel("\sigma");
%ylim([-3 3])

subplot(131);
plot(epsilon,epsilon_p)   ;
%ylim([-3 3])
xlabel("\epsilon");ylabel("\epsilon_p");
subplot(133);
plot(epsilon_p,sigma)   ;
xlabel("\epsilon_p");ylabel("\sigma")
%ylim([-3 3])         %limit y-axis
```

2.2 Charhe-décharge en traction

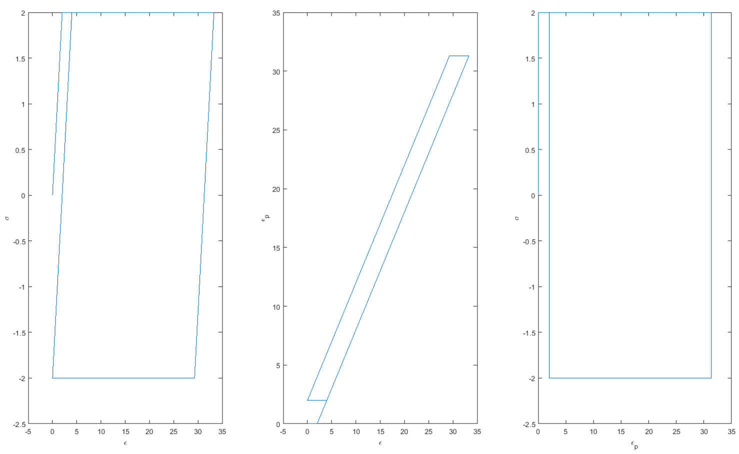


Les codes sont suivants :

```
function d_epsilon=charge_decharge(N)


d_epsilon=zeros(1,N);
d_epsilon(1)=0.1;

for i=2:N
if i<N/2
    d_epsilon(i)=d_epsilon(1);
else
    d_epsilon(i)= -d_epsilon(1);

end
end
end
```

Quand c'est charge,d$\epsilon > 0$. Par contre, quand c'est décharge, d$\epsilon < 0$

2.3 Un cycle de charge-décharge en traction-compression



Les codes sont suivants :

```
%un cycle de cherge-décharge
function d_epsilon=cycle_c(N)

d_epsilon=zeros(1,N);
d_epsilon(1)=0.1;
for  i=2:N
if i<N/3
  d_epsilon(i)=d_epsilon(1);
elseif  i>2*N/3
        d_epsilon(i)= d_epsilon(1);
 else
        d_epsilon(i)=-d_epsilon(1);

end
end
end
```

Par rapport de un cycle de charge-décharge et traction-compression, i'idée princcipale est de changer la valeur de $d\epsilon$. Donc dans le Matlab, j'ai utilisé trois phases de décrire cette situaion :
D'abord, $d\epsilon > 0$, ϵ_p=constant,ϵ_e monte, ϵ monte. Puis, ϵ_e =constant, ϵ_p monte,ϵ monte. En suite, ϵ_e descend, ϵ_p = constant,ϵ descend. A la fin, ϵ_p=constant,ϵ_e monte, ϵ monte.
Donc j'ai un cycle de (ϵ, σ).

2.4 Cycles de traction d'amplitude croissante, puis décroissante

Pour d'amplitude croissante et décrossante, je dois changer la valeur de $d\epsilon$ plus ou moins que la valeur précédente. Et je dois tracer les cycles dans le même graph. J'ai bien compris l'ideé, mais le problem pour moi est que le problem de Matlab parce que je ne peux tracer dans le même graph.

2.5 Cycles de traction-compression d'amplitude fixe, croissante, puis décroissante

Pour 2.5, l'idée est pariel.