

Exemple d'identification de paramètres pour un chargement de fluage

Données expérimentales

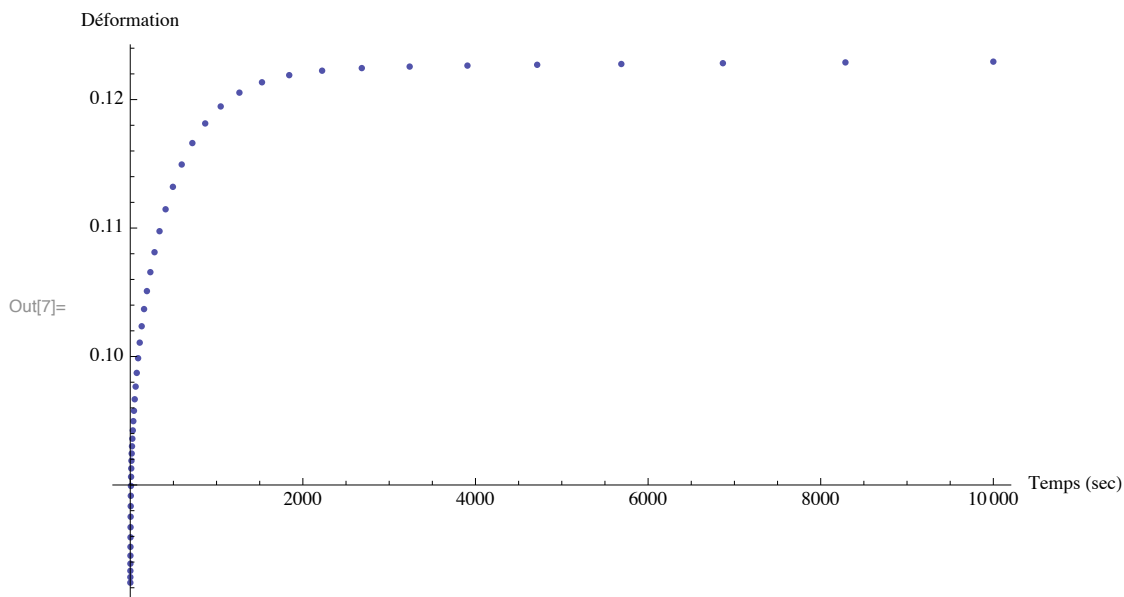
Importation des données expérimentales issues d'un essai de fluage à 20 MPa sur un matériau fictif

Ces étapes permettent de tracer la déformation axiale (eps1) et la déformation transverse (eps2) lors d'un essai de fluage en traction (on applique 20 MPa dans le sens axial et on mesure les déformations eps1 et eps2).

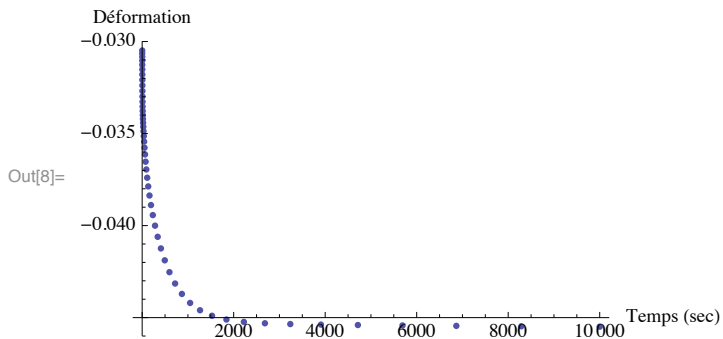
```
In[1]:= donnees = Import["data.tsv"];  
eps1 = Array[t, {50, 2}];  
eps2 = Array[t, {50, 2}];  
Do[eps1[[i, 1]] = donnees[[i, 1]];  
    eps1[[i, 2]] = donnees[[i, 2]]], {i, 1, 50}];  
Do[eps2[[i, 1]] = donnees[[i, 1]];  
    eps2[[i, 2]] = donnees[[i, 3]]], {i, 1, 50}];
```

Tracé des histoires de déformation (axe x = temps, axe y = déformation)

```
In[7]:= a3 = ListPlot[eps1, {PlotRange -> All, AxesLabel -> {"Temps (sec)", "Déformation"}}]
```



```
In[8]:= a4 = ListPlot[eps2, {PlotRange -> All, AxesLabel -> {"Temps (sec)", "Déformation"}}]
```



On peut donc voir que l'allure des courbes est bien celle d'un essai de fluage (extension dans le sens axial et contraction de Poisson dans le sens transverse).

Obtention des paramètres de la loi de comportement

Nous allons utiliser la méthodologie présentée aux transparents 75 à 78 pour obtenir les paramètres matériau

Détermination des λ_i

On se fixe a priori que l'on veut représenter la déformation de 1s à 10 000s. On va prendre 2 λ_i par décade de temps. Alors, si on distribue ces temps sur une échelle logarithmique, on aura:

$$\text{In}[10]:= \lambda_i = \left\{ 1, \frac{1}{10^{0.5}}, \frac{1}{10^1}, \frac{1}{10^{1.5}}, \frac{1}{10^2}, \frac{1}{10^{2.5}}, \frac{1}{10^3}, \frac{1}{10^{3.5}}, \frac{1}{10^4} \right\};$$

Définition des α_i et β_i

Comme on a 9 λ_i , on aura 9 α_i et β_i :

```
In[11]:=  $\alpha_i = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9\};$   

 $\beta_i = \{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8, \beta_9\};$ 
```

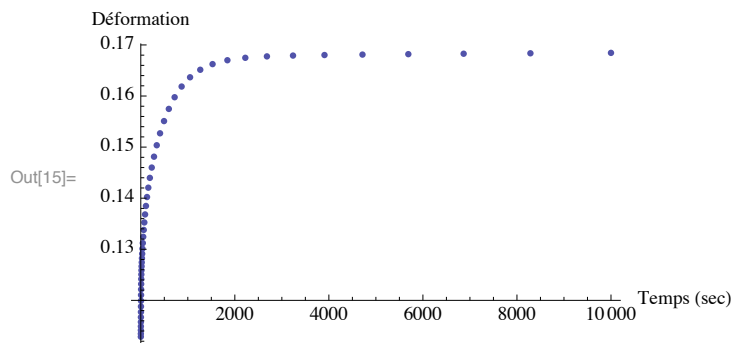
Définition de l'histoire qui permet de calculer les β_i (équation 95)

```
In[13]:= epsdagger1 = Array[t, {50, 2}];  

Do[epsdagger1[[i, 1]] = eps1[[i, 1]];  

  epsdagger1[[i, 2]] = eps1[[i, 2]] - eps2[[i, 2]]; {i, 1, 50}]
```

```
In[15]:= a1 = ListPlot[epsdagger1, {PlotRange -> All, AxesLabel -> {"Temps (sec)", "Déformation"}}]
```



Définition de la réponse théorique (équation 95)

Lorsque soumis à une histoire de fluage (contrainte de 20 MPa imposée soudainement et maintenue) l'équation (95) devient

$$\text{In[16]:= } \mathbf{epsdagger1theorique[t_]} = \mathbf{Expand}\left[\left(\beta_0 + \sum_{i=1}^9 (\beta_i [[i]] * (1 - \mathbf{Exp}[-\lambda_i [[i]] t]))\right) 20\right]$$

$$\text{Out[16]= } 20 \beta_0 + 20 \beta_1 - 20 e^{-t} \beta_1 + 20 \beta_2 - 20 e^{-0.316228 t} \beta_2 + 20 \beta_3 - 20 e^{-t/10} \beta_3 + \\ 20 \beta_4 - 20 e^{-0.0316228 t} \beta_4 + 20 \beta_5 - 20 e^{-t/100} \beta_5 + 20 \beta_6 - 20 e^{-0.00316228 t} \beta_6 + \\ 20 \beta_7 - 20 e^{-t/1000} \beta_7 + 20 \beta_8 - 20 e^{-0.000316228 t} \beta_8 + 20 \beta_9 - 20 e^{-t/10000} \beta_9$$

Définition du moindres carrés à résoudre (équation 97b)

$$\text{In[17]:= } \mathbf{mq1} = \mathbf{Expand}\left[\sum_{i=1}^{50} ((\mathbf{epsdagger1}[[i, 2]] - \mathbf{epsdagger1theorique}[\mathbf{epsdagger1}[[i, 1]]])^2)\right]$$

$$\text{Out[17]= } 1.01817 - 282.705 \beta_0 + 20000 \beta_0^2 - 276.5 \beta_1 + 39208.8 \beta_0 \beta_1 - \frac{800 \beta_0 \beta_1}{e} + 19287.5 \beta_1^2 + \frac{400 \beta_1^2}{e^2} - \frac{800 \beta_1^2}{e} - \\ 259.598 \beta_2 + 36015.7 \beta_0 \beta_2 + 35604.1 \beta_1 \beta_2 + 17030.4 \beta_2^2 - 234.707 \beta_3 + 32602.3 \beta_0 \beta_3 - \frac{800 \beta_0 \beta_3}{e^{1/10}} + \\ 32474.1 \beta_1 \beta_3 + \frac{800 \beta_1 \beta_3}{e^{11/10}} - \frac{800 \beta_1 \beta_3}{e} - \frac{800 \beta_1 \beta_3}{e^{1/10}} + 31038.2 \beta_2 \beta_3 + 15041.4 \beta_3^2 + \frac{400 \beta_3^2}{e^{1/5}} - \frac{800 \beta_3^2}{e^{1/10}} - \\ 205.681 \beta_4 + 27235.2 \beta_0 \beta_4 + 27182.6 \beta_1 \beta_4 + 26935.2 \beta_2 \beta_4 + 26181.9 \beta_3 \beta_4 + 12202.4 \beta_4^2 - 174.094 \beta_5 + \\ 23210.1 \beta_0 \beta_5 - \frac{800 \beta_0 \beta_5}{e^{1/100}} + 23196.1 \beta_1 \beta_5 + \frac{800 \beta_1 \beta_5}{e^{101/100}} - \frac{800 \beta_1 \beta_5}{e} - \frac{800 \beta_1 \beta_5}{e^{1/100}} + 22319.1 \beta_2 \beta_5 + \\ 22848.6 \beta_3 \beta_5 + \frac{800 \beta_3 \beta_5}{e^{11/100}} - \frac{800 \beta_3 \beta_5}{e^{1/10}} - \frac{800 \beta_3 \beta_5}{e^{1/100}} + 21286.7 \beta_4 \beta_5 + 10153.2 \beta_5^2 + \frac{400 \beta_5^2}{e^{1/50}} - \frac{800 \beta_5^2}{e^{1/100}} - \\ 140.026 \beta_6 + 17544.5 \beta_0 \beta_6 + 17539.1 \beta_1 \beta_6 + 17512.7 \beta_2 \beta_6 + 17423.6 \beta_3 \beta_6 + 17150.9 \beta_4 \beta_6 + \\ 16387.2 \beta_5 \beta_6 + 7303.28 \beta_6^2 - 103.199 \beta_7 + 13452. \beta_0 \beta_7 - \frac{800 \beta_0 \beta_7}{e^{1/1000}} + 13450.6 \beta_1 \beta_7 + \frac{800 \beta_1 \beta_7}{e^{1001/1000}} - \\ \frac{800 \beta_1 \beta_7}{e} - \frac{800 \beta_1 \beta_7}{e^{1/1000}} + 12642.7 \beta_2 \beta_7 + 13414.1 \beta_3 \beta_7 + \frac{800 \beta_3 \beta_7}{e^{101/1000}} - \frac{800 \beta_3 \beta_7}{e^{1/10}} - \frac{800 \beta_3 \beta_7}{e^{1/1000}} + \\ 12524.1 \beta_4 \beta_7 + 13051. \beta_5 \beta_7 + \frac{800 \beta_5 \beta_7}{e^{11/1000}} - \frac{800 \beta_5 \beta_7}{e^{1/100}} - \frac{800 \beta_5 \beta_7}{e^{1/1000}} + 11487.2 \beta_6 \beta_7 + 5253.3 \beta_7^2 + \frac{400 \beta_7^2}{e^{1/500}} - \\ \frac{800 \beta_7^2}{e^{1/1000}} - 64.3133 \beta_8 + 7785.42 \beta_0 \beta_8 + 7784.88 \beta_1 \beta_8 + 7782.23 \beta_2 \beta_8 + 7773.16 \beta_3 \beta_8 + 7744.28 \beta_4 \beta_8 + \\ 7654.14 \beta_5 \beta_8 + 7381. \beta_6 \beta_8 + 6617.21 \beta_7 \beta_8 + 2433.02 \beta_8^2 - 30.2919 \beta_9 + 4447.39 \beta_0 \beta_9 - \frac{800 \beta_0 \beta_9}{e^{1/10000}} + \\ 4447.25 \beta_1 \beta_9 + \frac{800 \beta_1 \beta_9}{e^{10001/10000}} - \frac{800 \beta_1 \beta_9}{e} - \frac{800 \beta_1 \beta_9}{e^{1/10000}} + 3646.46 \beta_2 \beta_9 + 4443.58 \beta_3 \beta_9 + \frac{800 \beta_3 \beta_9}{e^{1001/10000}} - \\ \frac{800 \beta_3 \beta_9}{e^{1/10}} - \frac{800 \beta_3 \beta_9}{e^{1/10000}} + 3634.41 \beta_4 \beta_9 + 4405.51 \beta_5 \beta_9 + \frac{800 \beta_5 \beta_9}{e^{101/10000}} - \frac{800 \beta_5 \beta_9}{e^{1/100}} - \frac{800 \beta_5 \beta_9}{e^{1/10000}} + 3515.35 \beta_6 \beta_9 + \\ 4042.21 \beta_7 \beta_9 + \frac{800 \beta_7 \beta_9}{e^{11/10000}} - \frac{800 \beta_7 \beta_9}{e^{1/1000}} - \frac{800 \beta_7 \beta_9}{e^{1/10000}} + 2500.58 \beta_8 \beta_9 + 1065.78 \beta_9^2 + \frac{400 \beta_9^2}{e^{1/5000}} - \frac{800 \beta_9^2}{e^{1/10000}}$$

On doit donc chercher les valeurs de β_i qui minimisent ce moindres carrés. Cela peut se faire avec la fonction `NMinimize` de Mathematica

Détermination des β_i

```
In[18]:= NMinimize[mq1, {β0, β1, β2, β3, β4, β5, β6, β7, β8, β9}]
```

```
Out[18]= {7.58067 × 10-7,
  {β0 → 0.00553639, β1 → -0.000162872, β2 → 0.000736038, β3 → -8.85237 × 10-6, β4 → 0.00033832,
  β5 → 0.000108325, β6 → 0.00109308, β7 → 0.00109183, β8 → -0.000682337, β9 → 0.000558725}}
```

On peut voir ici que l'on a obtenu des β_i négatifs, ce qui n'est pas acceptable thermodynamiquement. Essayons avec le changement de variables proposé au transparent 77

Détermination des β_i acceptables thermodynamiquement

```
In[19]:= βi2 = {y1, y2, y3, y4, y5, y6, y7, y8, y9}^2;
```

$$\text{epsdagger1theorique}[t_]=\text{Expand}\left[\left(y_0^2+\sum_{i=1}^9(\beta i2[[i]]*(1-\text{Exp}[-\lambda i[[i]]t]))\right)20\right]$$

```
Out[20]= 20 y0^2 + 20 y1^2 - 20 e-t y1^2 + 20 y2^2 - 20 e-0.316228 t y2^2 + 20 y3^2 - 20 e-t/10 y3^2 +
  20 y4^2 - 20 e-0.0316228 t y4^2 + 20 y5^2 - 20 e-t/100 y5^2 + 20 y6^2 - 20 e-0.00316228 t y6^2 +
  20 y7^2 - 20 e-t/1000 y7^2 + 20 y8^2 - 20 e-0.000316228 t y8^2 + 20 y9^2 - 20 e-t/10000 y9^2
```

```
In[21]:= mq1 = Expand[Sum[(epsdagger1[[i, 2]] - epsdagger1theorique[epsdagger1[[i, 1]]])^2, {i, 1, 9}]];
```

```
In[22]:= res1 = NMinimize[mq1, {y0, y1, y2, y3, y4, y5, y6, y7, y8, y9}]
```

```
Out[22]= {2.38098 × 10-6,
  {y0 → -0.0738238, y1 → 4.18888 × 10-7, y2 → -0.0246532, y3 → 0.0080578, y4 → -0.0183172,
  y5 → -1.02869 × 10-6, y6 → 0.0370856, y7 → -0.0243136, y8 → -2.37655 × 10-8, y9 → -2.14629 × 10-7}}
```

On trouve des y_i qui sont négatifs, mais avec le changement de variable $\beta_i = y_i^2$, on obtiendra des β_i qui sont tous positifs

```
In[23]:= Do[βi[[i]] = res1[[2, i + 1, 2]]^2, {i, 1, 9}];
  β0 = res1[[2, 1, 2]]^2
```

```
Out[24]= 0.00544995
```

(la valeur en haut est celle de β_0)

```
In[25]:= βi
```

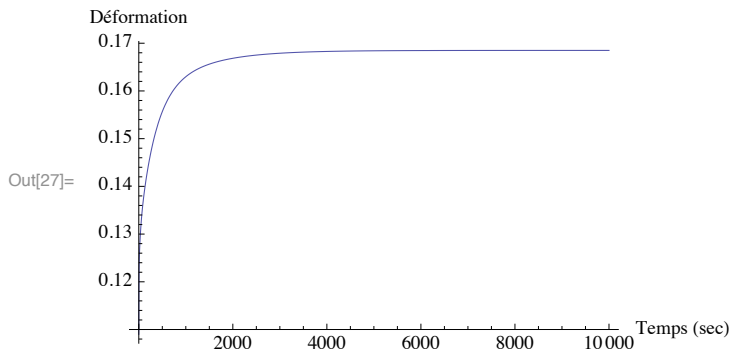
```
Out[25]= {1.75467 × 10-13, 0.000607783, 0.0000649282, 0.000335519,
  1.05819 × 10-12, 0.00137534, 0.000591152, 5.64797 × 10-16, 4.60654 × 10-14}
```

On a donc atteint notre objectif d'avoir des β_i positifs. Passons maintenant au cas des α_i .

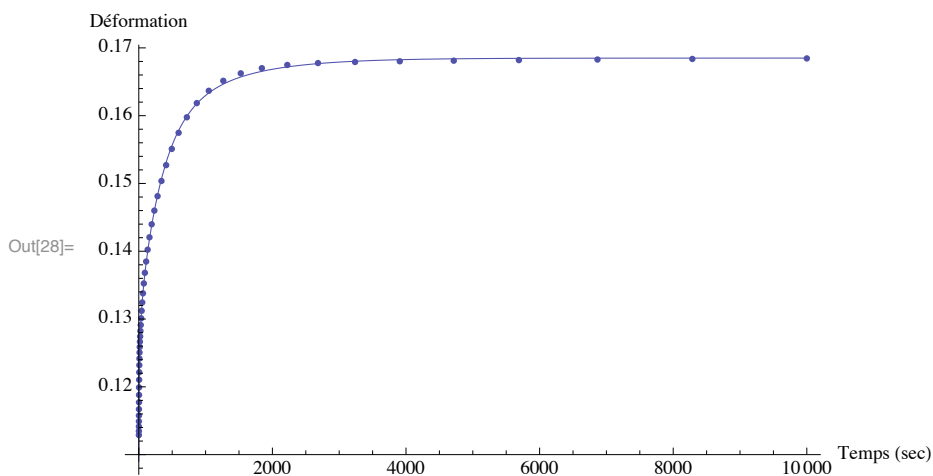
Comparaison entre les données expérimentales et théoriques pour `epsdagger1`

```
In[26]:= epsdagger1theorique[t_] = Expand[β0 + Sum[βi[[i]]*(1 - Exp[-λi[[i]]t]), {i, 1, 9}]]20;
```

```
In[27]:= b1 = Plot[epsdagger1theorique[t], {t, 0, 10 000},
  {PlotRange -> All, AxesLabel -> {"Temps (sec)", "Déformation"}}]
```



```
In[28]:= Show[a1, b1]
```

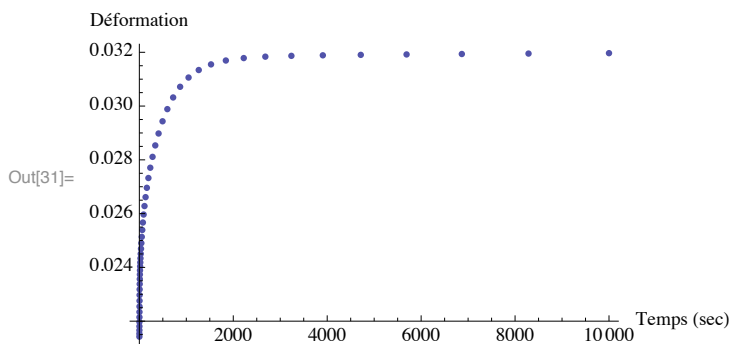


On peut donc voir que notre approximation est assez bonne par rapport aux données expérimentales

Définition de l'histoire qui permet de calculer les α_i (équation 96)

```
In[29]:= epsdagger2 = Array[t, {50, 2}];
Do[epsdagger2[[i, 1]] = eps1[[i, 1]];
  epsdagger2[[i, 2]] = eps1[[i, 2]] + 2 * eps2[[i, 2]]; {i, 1, 50}]
```

```
In[31]:= a2 = ListPlot[epsdagger2, {PlotRange -> All, AxesLabel -> {"Temps (sec)", "Déformation"}}]
```



Définition de la réponse théorique (équation 95)

Lorsque soumis à une histoire de fluage (contrainte de 20 MPa imposée soudainement et maintenue) l'équation (96) devient

$$\text{In[32]:= } \text{epsdagger2theorique}[t_] = \text{Expand}\left[\left(\alpha_0 + \sum_{i=1}^9 (\alpha_i [[i]] * (1 - \text{Exp}[-\lambda_i [[i]] t]))\right) 20\right]$$

$$\text{Out[32]= } 20 \alpha_0 + 20 \alpha_1 - 20 e^{-t} \alpha_1 + 20 \alpha_2 - 20 e^{-0.316228 t} \alpha_2 + 20 \alpha_3 - 20 e^{-t/10} \alpha_3 +$$

$$20 \alpha_4 - 20 e^{-0.0316228 t} \alpha_4 + 20 \alpha_5 - 20 e^{-t/100} \alpha_5 + 20 \alpha_6 - 20 e^{-0.00316228 t} \alpha_6 +$$

$$20 \alpha_7 - 20 e^{-t/1000} \alpha_7 + 20 \alpha_8 - 20 e^{-0.000316228 t} \alpha_8 + 20 \alpha_9 - 20 e^{-t/10000} \alpha_9$$

Définition du moindres carrés à résoudre (équation 97b)

$$\text{In[33]:= } \text{mq2} = \text{Expand}\left[\sum_{i=1}^{50} ((\text{epsdagger2}[[i, 2]] - \text{epsdagger2theorique}[\text{epsdagger2}[[i, 1]]])^2)\right];$$

On doit donc chercher les valeurs de α_i qui minimisent ce moindres carrés. Cela peut se faire avec la fonction NMinimize de Mathematica

Détermination des α_i

$$\text{In[34]:= } \text{NMinimize}[\text{mq2}, \{\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9\}]$$

$$\text{Out[34]= } \{2.73032 \times 10^{-8},$$

$$\{\alpha_0 \rightarrow 0.0010507, \alpha_1 \rightarrow -0.00003091, \alpha_2 \rightarrow 0.000139686, \alpha_3 \rightarrow -1.68001 \times 10^{-6}, \alpha_4 \rightarrow 0.0000642067,$$

$$\alpha_5 \rightarrow 0.0000205581, \alpha_6 \rightarrow 0.000207445, \alpha_7 \rightarrow 0.000207209, \alpha_8 \rightarrow -0.000129495, \alpha_9 \rightarrow 0.000106035\}\}$$

On peut voir ici que l'on a obtenu des α_i négatifs, ce qui n'est pas acceptable thermodynamiquement. Essayons avec le changement de variables proposé au transparent 77

Détermination des α_i acceptables thermodynamiquement

$$\text{In[35]:= } \alpha_i2 = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}^2;$$

$$\text{epsdagger2theorique}[t_] = \text{Expand}\left[\left(x_0^2 + \sum_{i=1}^9 (\alpha_i2 [[i]] * (1 - \text{Exp}[-\lambda_i [[i]] t]))\right) 20\right]$$

$$\text{Out[36]= } 20 x_0^2 + 20 x_1^2 - 20 e^{-t} x_1^2 + 20 x_2^2 - 20 e^{-0.316228 t} x_2^2 + 20 x_3^2 - 20 e^{-t/10} x_3^2 +$$

$$20 x_4^2 - 20 e^{-0.0316228 t} x_4^2 + 20 x_5^2 - 20 e^{-t/100} x_5^2 + 20 x_6^2 - 20 e^{-0.00316228 t} x_6^2 +$$

$$20 x_7^2 - 20 e^{-t/1000} x_7^2 + 20 x_8^2 - 20 e^{-0.000316228 t} x_8^2 + 20 x_9^2 - 20 e^{-t/10000} x_9^2$$

$$\text{In[37]:= } \text{mq2} = \text{Expand}\left[\sum_{i=1}^{50} ((\text{epsdagger2}[[i, 2]] - \text{epsdagger2theorique}[\text{epsdagger2}[[i, 1]]])^2)\right];$$

$$\text{In[38]:= } \text{res2} = \text{NMinimize}[\text{mq2}, \{x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}]$$

$$\text{Out[38]= } \{8.57552 \times 10^{-8},$$

$$\{x_0 \rightarrow 0.0321605, x_1 \rightarrow 3.28576 \times 10^{-8}, x_2 \rightarrow 0.0107399, x_3 \rightarrow -0.00351024, x_4 \rightarrow 0.00797969,$$

$$x_5 \rightarrow 2.19963 \times 10^{-8}, x_6 \rightarrow -0.0161559, x_7 \rightarrow -0.010592, x_8 \rightarrow -2.27461 \times 10^{-8}, x_9 \rightarrow 8.98566 \times 10^{-9}\}\}$$

On trouve des y_i qui sont négatifs, mais avec le changement de variable $\alpha_i = x_i^2$, on obtiendra des α_i qui sont tous positifs

```
In[39]:= Do[ $\alpha_i[[i]] = \text{res2}[[2, i + 1, 2]]^2, \{i, 1, 9\}];$   

 $\alpha_0 = \text{res2}[[2, 1, 2]]^2$ 
```

```
Out[40]= 0.0010343
```

(la valeur en haut est celle de α_0)

```
In[41]:=  $\alpha_i$ 
```

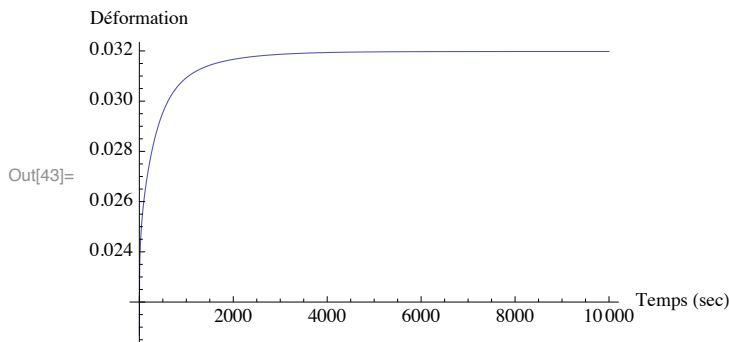
```
Out[41]= {1.07962  $\times 10^{-15}$ , 0.000115346, 0.0000123218, 0.0000636754,  

4.83839  $\times 10^{-16}$ , 0.000261013, 0.00011219, 5.17385  $\times 10^{-16}$ , 8.0742  $\times 10^{-17}$ }
```

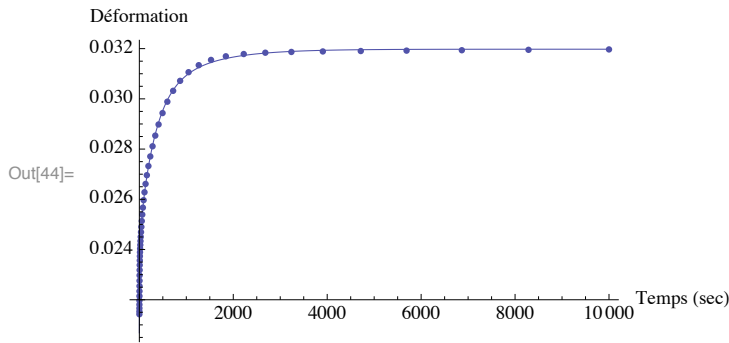
```
In[42]:= epsdagger2theorique[t_] = Expand[ $\left(\alpha_0 + \sum_{i=1}^9 (\alpha_i[[i]] * (1 - \text{Exp}[-\lambda_i[[i]] t])\right) 20$ ];
```

```
In[43]:= b2 = Plot[epsdagger2theorique[t], {t, 0, 10000},  

{PlotRange -> All, AxesLabel -> {"Temps (sec)", "Déformation"}}]
```



```
In[44]:= Show[a2, b2]
```



On peut aussi voir que l'on colle assez bien aux données expérimentales

Vérification que l'on colle bien aux données de l'essai

Ce que l'on a fait jusqu'à maintenant nous a permis de trouver les paramètres α_i et β_i qui collent aux données epsdagger1 et epsdagger2. Voyons si les paramètres obtenus font en sorte que l'on retrouve bien les données expérimentales de l'essai. Pour ce faire, on va utiliser la décomposition tensorielle avec les projecteurs (transparents 13 et 14)

```
In[45]:=  $\mathbf{J} = \frac{1}{3.} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}; \mathbf{K} = \frac{1}{3.} \begin{pmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix};$ 
```

```
In[46]:= souplessefluage[t_] = ( $\alpha_0$  J +  $\beta_0$  K) +  $\sum_{i=1}^9 ((\alpha_i[[i]]$  J +  $\beta_i[[i]]$  K) * (1 - Exp[- $\lambda_i[[i]]$  t]));
```

Ce résultat nous donne la souplesse de fluage - tenseur d'ordre 4 isotrope - de notre matériau. Si on lui applique l'histoire de contrainte suivante

```
In[47]:=  $\sigma = \begin{pmatrix} 20. \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix};$ 
```

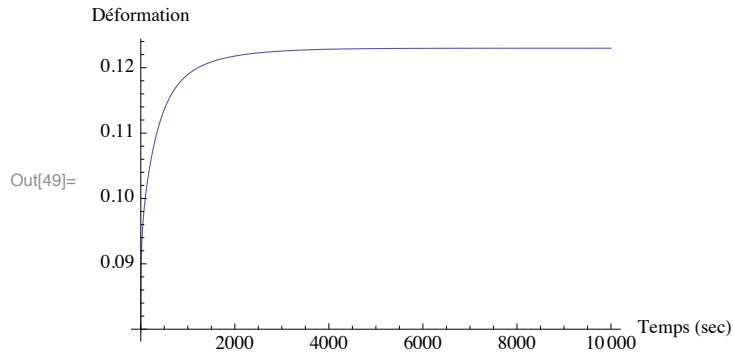
on obtient:

```
In[48]:= reponse = Expand[souplessefluage[t]. $\sigma$ ]
```

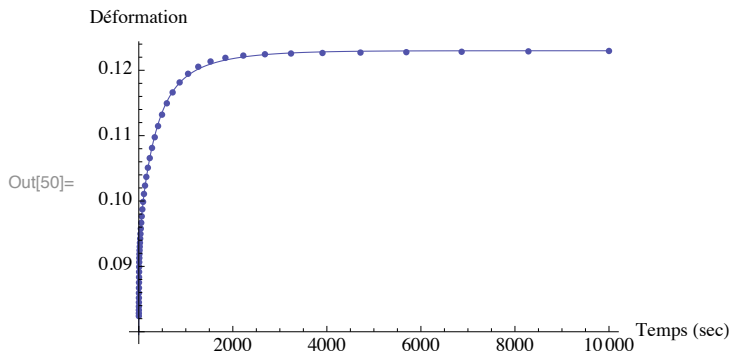
```
Out[48]= { { 0.122988 - 2.34676  $\times 10^{-12}$  e-t - 0.00887274 e-0.316228 t - 0.000947854 e-t/10 -  
0.00489809 e-0.0316228 t - 1.41125  $\times 10^{-11}$  e-t/100 - 0.020078 e-0.00316228 t -  
0.00862996 e-t/1000 - 1.09799  $\times 10^{-14}$  e-0.000316228 t - 6.14744  $\times 10^{-13}$  e-t/10000 },  
 { -0.0455055 + 1.16258  $\times 10^{-12}$  e-t + 0.00328291 e-0.316228 t + 0.000350709 e-t/10 +  
0.00181229 e-0.0316228 t + 7.0514  $\times 10^{-12}$  e-t/100 + 0.00742885 e-0.00316228 t +  
0.00319309 e-t/1000 + 3.16081  $\times 10^{-16}$  e-0.000316228 t + 3.06565  $\times 10^{-13}$  e-t/10000 },  
 { -0.0455055 + 1.16258  $\times 10^{-12}$  e-t + 0.00328291 e-0.316228 t + 0.000350709 e-t/10 +  
0.00181229 e-0.0316228 t + 7.0514  $\times 10^{-12}$  e-t/100 + 0.00742885 e-0.00316228 t +  
0.00319309 e-t/1000 + 3.16081  $\times 10^{-16}$  e-0.000316228 t + 3.06565  $\times 10^{-13}$  e-t/10000 }, {0}, {0}, {0} }
```

Le premier terme de ce vecteur est la déformation dans le sens de l'essai. Comparons avec les données expérimentales

```
In[49]:= b3 = Plot[reponse[[1, 1]], {t, 0, 10000},  
 {PlotRange → All, AxesLabel → {"Temps (sec)", "Déformation"}}
```

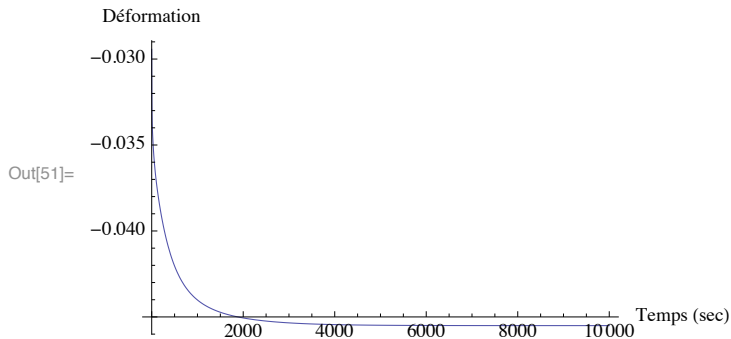


In[50]:= **Show[a3, b3]** (*a3 est le graphe pour la déformation axiale obtenu au début du fichier*)

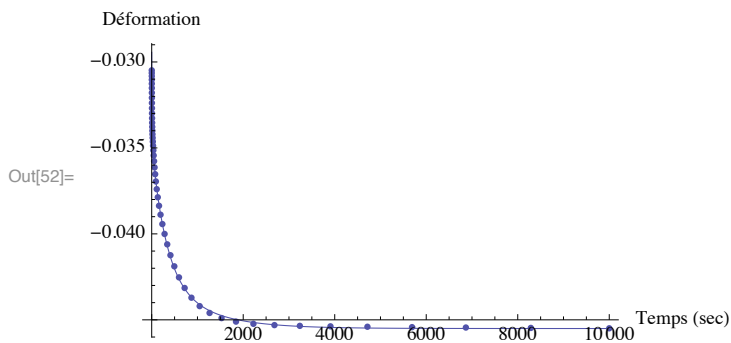


On voit que l'on retrouve bien la déformation axiale. Comparons pour la déformation transverse:

In[51]:= **b4 = Plot[reponse[[2, 1]], {t, 0, 10 000},**
{PlotRange -> All, AxesLabel -> {"Temps (sec)", "Déformation"}}]



In[52]:= **Show[a4, b4]** (*a4 est le graphe pour la déformation transverse obtenu au début du fichier*)



Encore une fois, on voit que l'on colle très bien aux données.