Fonctions à programmer

Semaine 5

Pour cette semaine, vous devez écrire un programme qui permet de trouver les paramètres d'une loi de comportement viscoélastique linéaire à partir des données expérimentales.

1 Essai de fluage-recouvrance idéalisé

1.1 Équations à optimiser

Soit un matériau viscoélastique linéaire soumis à une contrainte uni-axiale $\sigma(t) = \sigma_0 H(t)$. Si on note respectivement par $\check{\varepsilon}_{11}(t)$ et $\check{\varepsilon}_{22}(t)$ la déformation axiale et transverse, on a que :

$$\check{\varepsilon}^{\dagger}(t) = \check{\varepsilon}_{11}(t) - \check{\varepsilon}_{22}(t) = \beta_0\sigma_0 + \sum \beta_i\big(1 - e^{-\lambda_i t}\big)\sigma_0 \tag{1a}$$

$$\check{\varepsilon}^{\dagger\dagger}(t) = \check{\varepsilon}_{11}(t) + 2\check{\varepsilon}_{22}(t) = \alpha_0\sigma_0 + \sum \alpha_i\big(1 - e^{-\lambda_i t}\big)\sigma_0 \tag{1b}$$

Il faudra donc minimiser les quantités suivantes :

$$\kappa_1 = \sum \left[\check{\varepsilon}^{\dagger\dagger}(t) - \hat{\varepsilon}^{\dagger\dagger}(t)\right]^2 \tag{2a}$$

$$\kappa_2 = \sum \left[\check{\varepsilon}^{\dagger}(t) - \hat{\varepsilon}^{\dagger}(t)\right]^2 \tag{2b}$$

avec

$$\hat{\varepsilon}^{\dagger}(t) = \hat{\varepsilon}_{11}(t) - \hat{\varepsilon}_{22}(t) \tag{3a}$$

$$\hat{\varepsilon}^{\dagger\dagger}(t) = \hat{\varepsilon}_{11}(t) + 2\hat{\varepsilon}_{22}(t) \tag{3b}$$

où $\hat{\varepsilon}_{11}(t)$ et $\hat{\varepsilon}_{22}(t)$ représentent respectivement la déformation axiale et transverse obtenues expérimentalement.

1.2 Fonctions à programmer

- Une fonction qui lit les données expérimentales à partir d'un fichier Excel. Le fichier Excel aura toujours le même format. Les 3 premières colonnes représentent respectivement le temps, la déformation axiale $\hat{\varepsilon}_{11}(t)$ et la déformation transverse $\hat{\varepsilon}_{22}(t)$. Vous pouvez utiliser la fonction `xlsread` de MATLAB.
- Une fonction qui prend comme arguments un vecteur contenant les α_i, un vecteur contenant les pas de temps, un vecteur contenant les déformations axiales expérimentales, un vecteur contenant les déformations transverses expérimentales, un vecteur contenant les λ_i sélectionnés et un scalaire représentant la contrainte imposée σ_0. La fonction doit

retourner un vecteur de résidus. Chaque terme de ce vecteur de résidus représente la différence entre $\breve{\varepsilon}^{\dagger\dagger}(t_n)$ et $\hat{\varepsilon}^{\dagger\dagger}(t_n)$ pour un temps donné t_n. Le vecteur de résidus aura donc la même taille que celui contenant les pas de temps.

- Une fonction qui prend comme arguments un vecteur contenant les β_i, un vecteur contenant les pas de temps, un vecteur contenant les déformations axiales expérimentales, un vecteur contenant les déformations transverses expérimentales, un vecteur contenant les λ_i sélectionnés et un scalaire représentant la contrainte imposée σ_0. La fonction doit retourner un vecteur de résidus. Chaque terme de ce vecteur de résidus représente la différence entre $\breve{\varepsilon}^{\dagger}(t_n)$ et $\hat{\varepsilon}^{\dagger}(t_n)$ pour un temps donné t_n. Le vecteur de résidus aura donc la même taille que celui contenant les pas de temps.

1.3 Programme Principal

Votre programme principal doit :

- Lire les résultats expérimentaux.
- Définir la contrainte appliquée σ_0.
- Définir les λ_i.
- Initialiser les α_i et β_i.
- Calculer les α_i et β_i optimaux. Vous devez utiliser la fonction `lsqnonlin` de MATLAB, une fois pour les α_i et une fois pour β_i. La fonction `lsqnonlin` devra appeler les fonctions que vous avez codées à la Section 1.2. Vous pouvez spécifier des options pour la fonction `lsqnonlin` dont le type d'algorithme (utilisez l'algorithme de LevenBerg-Marquardt), la tolérance, le nombre maximal d'itérations, etc.
- Négliger les α_i et β_i qui sont très faibles par rapport aux autres.
- Tracer les déformations théoriques $\breve{\varepsilon}_{11}(t)$ et $\breve{\varepsilon}_{22}(t)$. Comparez les courbes théoriques avec les courbes expérimentales. Ceci vous permettra de valider vos résultats.

2 Essai de fluage-recouvrance réel

Le programme que vous avez développé à la Section 1 est suffisant pour résoudre les exercices suggérés ainsi que l'exercice que vous aurez à l'Intra 1. Toutefois, pour le laboratoire, ça se complique un petit peu.

Vous devez tenir compte de deux modifications majeures :

- La contrainte appliquée n'est pas idéalisée. Elle prend la forme d'une pente positive entre 0 et t_1, d'une constante entre t_1 et t_2 et d'une pente négative entre t_2 et t_3 (où t_1, t_2 et t_3 seront à déterminer). Ainsi, on ne peut plus écrire $\sigma(t) = \sigma_0 H(t)$. Les Équations (1a) et (1b) ne sont donc plus valables. Vous devez les modifier.
- Vous devez optimiser <u>simultanément</u> plusieurs courbes expérimentales. Vous devez donc modifier les fonctions que vous avez codées à la Section 1.2. Les vecteurs de résidus

doivent maintenant contenir les résidus provenant de toutes les courbes expérimentales.