IDENTIFICATION DE MODELES RHEOLOGIQUES PAR PROGRAMMATION GENETIQUE

M. Sebag and M. Schoenauer and H. Maitournam
LMS-CNRS URA 317 CMAP-CNRS URA 756
Ecole Polytechnique
91128 Palaiseau Cedex
{ Michele.Sebag, Marc.Schoenauer, Habibou.Maitournam }@polytechnique.fr

Résumé : Le travail présenté concerne l'identification automatique de lois de comportement à partir des données expérimentales disponibles, restreinte au cas de lois de comportement élasto-visco-plastiques uni-dimensionnelles décrites sous forme de modèles rhéologiques.
La programmation génétique, originellement conçue pour l'identification de fonctions ou de programmes optimisant un objectif donné, est adaptée à l'identification automatique de *l'architecture et des coefficients* d'un modèle rhéologique à partir d'expériences.
Cette approche est validée numériquement par l'identification d'un modèle rhéologique (proposé pour le polyethylène) à partir de son comportement simulé.

Abstract: Automatic identification of macro-mechanical models is obtained by adapting genetic programming techniques. The search space is that of rheological models, described as combinations of elements (springs, sliders, and dashpots) connected in series and/or in parallel.
This approach is validated by identifying a given rheologic model (proposed for polyethylene) from its simulated behavior under prescribed sollicitations.

1. Introduction

Le calcul de structures requiert de disposer des lois de comportement des matériaux considérés. La précision des modèles macro-mécaniques commande la finesse des simulations possibles, et par contre-coup, la qualité de la conception et le coût de réalisation. Or, des matériaux nouveaux tels que composites ou polymères interviennent souvent dans les structures sans que leur loi de comportement soit identifiée avec la précision souhaitable dans la gamme des sollicitations envisagées. La modélisation macro-mécanique apparaît donc comme un enjeu d'importance majeure tant scientifique qu'industriel. Les approches existantes reposent essentiellement sur les connaissances, l'intuition et le *sens de la mécanique* du chercheur ou de l'ingénieur. Celui-ci peut proposer un modèle de comportement en s'appuyant sur les analogies entre le matériau considéré et un matériau de référence, ou en procédant empiriquement. Les coefficients de la loi proposée sont ensuite ajustés en fonction des courbes expérimentales disponibles. Il est également possible de procéder à une analyse au niveau micro-mécanique. Le chercheur propose alors un modèle micro-mécanique qui débouchera analytiquement à terme sur une loi macro-mécanique du matériau [SP 87].
Ces approches peuvent conduire à une impasse lorsque le comportement non-linéaire du matériau échappe aux modèles imaginés, ou lorsque la loi macro-mécanique construite e.g. par homogénéisation d'un modèle micro-mécanique est trop complexe pour permettre des simulations abordables.

Cet article concerne l'identification automatique de la loi de comportement d'un matériau à partir des résultats d'expérimentation disponibles. Nous nous restreignons ici à l'identification de lois élasto-visco-plastiques unidimensionnelles, représentées sous forme de modèles rhéologiques connectant en série ou en parallèle des ressorts, des patins et des amortisseurs, qui correspondent respectivement à des comportements purement élastiques, plastiques et visqueux [PER 60]. Cet objectif constitue l'étape élémentaire de l'identification d'un modèle tri-dimensionnel en mécanique des solides [KIC 96] ; il répond aussi directement à certains besoins en bio-mécanique (modélisation de ligaments). De façon précise, le but est d'identifier à la fois *l'architecture* du modèle rhéologique, et les *coefficients numériques* de ce modèle. La difficulté est donc double : il faut déterminer le nombre d'éléments du modèle et leur agencement, puis les coefficients optimaux correspondants. Or, même en supposant le modèle donné, la détermination des coefficients optimaux est difficile [BOY 95] : elle correspond dans le cas général à l'optimisation d'une fonction non convexe.

L'approche utilisée est inspirée des algorithmes génétiques (AGs) [GOL 89], qui sont connus comme capables de répondre à des problèmes d'optimisation mal posés. La programmation génétique [KOZ 94] étend cette approche au cas d'espaces de fonctions, et nous en rappellerons brièvement le principe dans la partie 2.

La programmation génétique a été adaptée à l'exploration et à la manipulation de modèles rhéologiques (partie 3). Pour pouvoir évaluer un modèle rhéologique, il faut pouvoir déterminer et simuler la loi de comportement correspondante : la résolution du problème inverse (identification) se ramène au problème direct (simulation).

La faisabilité de l'approche est établie numériquement, en retrouvant un modèle complexe (proposé pour le polyethylène [KIC 96]) à partir de courbes "expérimentales" obtenues par simulation de ce modèle (partie 4).

2. Algorithmes génétiques et Programmation génétique

Cette partie décrit brièvement les méthodes d'optimisation par évolution (voir notamment [GOL 89] pour une présentation détaillée). Etant donné une fonction $\mathcal{F}$ définie sur un espace Ω à valeurs réelles, le but est de déterminer le ou les optima de $\mathcal{F}$:

$$Trouver\ x\ /\ \mathcal{F}(x) = Max\{\mathcal{F}(y), y \in \Omega\}$$

L'optimisation par évolution se fonde sur une analogie avec l'évolution darwinienne : étant donné une population d'individus (points de Ω) qui se reproduisent par mutation et croisement, seuls les individus les plus aptes survivent au bout d'un nombre donné de générations ; et si l'aptitude d'un individu x est donnée par $\mathcal{F}(x)$, les "plus aptes" sont exactement les optima de $\mathcal{F}$. Un algorithme d'évolution est donc défini par les étapes suivantes :

1. Génération (en général aléatoire) de la population initiale Π_0.
2. Evaluation : pour tout x de la population courante Π_i, calcul de $\mathcal{F}(x)$.
3. Sélection des individus les meilleurs de Π_i (sélection déterministe ou non) ;
4. Mutation-Croisement : les individus sélectionnés (parents) sont mutés ou croisés pour obtenir des individus nouveaux (enfants), qui formeront la population Π_{i+1}. Retour à l'étape 2 en incrémentant i.

Cette approche ne dépend d'aucune hypothèse (convexité, dérivabilité, continuité) sur la fonction $\mathcal{F}$ à optimiser. $\mathcal{F}$ peut même être donnée de façon uniquement calculable (i.e. comme résultat d'un algorithme).

La programmation génétique [KOZ 94] étend l'approche ci-dessus au cas où l'espace de recherche Ω est un espace de fonctions décrites par un ensemble d'opérandes $\mathcal{T}$ et d'opérateurs $\mathcal{N}$. Pour prendre un exemple, l'ensemble des polynomes de degré quelconque à 2 variables est décrit par l'ensemble des opérateurs $\mathcal{N} = \{+, \times\}$ et l'ensemble des opérandes $\mathcal{T} = \{X, Y, \mathcal{R}\}$ où X et Y désignent les variables du polynome et $\mathcal{R}$ l'ensemble des constantes réelles.

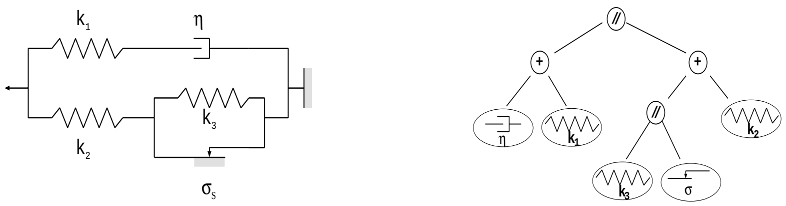
Ω correspond alors à l'ensemble des arbres dont les feuilles (noeuds terminaux) sont dans $\mathcal{T}$ et les noeuds internes sont dans $\mathcal{N}$. Des procédures ad hoc permettant d'initialiser, de muter et de croiser des arbres ont été définies [KOZ 94]. Ainsi, la mutation consiste à remplacer un sous-arbre dont la position est choisie aléatoirement, par un nouveau sous-arbre aléatoire. Et les "enfants" issus du croisement de deux arbres parents sont obtenus en échangeant deux sous-arbres des parents.

3. Identification de modèles rhéologiques

L'approche de la programmation génétique décrite ci-dessus est adaptée à l'exploration et à l'identification de modèles rhéologiques.

3.1 L'espace des modèles rhéologiques

Rappelons que nous nous sommes limités à l'identification de modèles rhéologiques décrits par des connexions en série ou en parallèle de trois types d'éléments, les ressorts, les patins et les amortisseurs. Un exemple est donné en Figure 1 (a).



(a) Modèle rhéologique (b) Arbre rhéologique

Figure 1: Modélisation rhéologique du polyéthylène [KIC 96].

Un modèle rhéologique peut de façon naturelle être représenté par un arbre (Fig 1 (b)) : les opérateurs sont ici les connecteurs série ou parallèle ($\mathcal{N} = \{\Sigma, ll\}$) et les opérandes sont les trois éléments ressorts, patins et amortisseurs; un opérande est ainsi décrit par son type et par une valeur numérique (la raideur k d'un ressort, le seuil σ_S d'un patin, et la viscosité η d'un amortisseur : $\mathcal{T} = \{k, \sigma_S, \eta\}$).

Les procédures générales d'initialisation, de croisement et de mutation définies sur les arbres permettent ainsi de générer et de manipuler tous les modèles rhéologiques de l'espace de recherche considéré.

3.2 Performance d'un modèle

Il reste à poser l'identification comme un problème d'optimisation. Clairement, le but consiste à minimiser la fonction $\mathcal{F}$ d'erreur de modélisation, calculée comme la différence (relative ou absolue, en norme L_∞ ou en norme L_2) entre le comportement simulé d'un modèle, et le comportement expérimental du matériau à modéliser.

Dans le cas d'une expérience pilotée en déformation, auquel nous nous limitons ici, une expérience $\mathcal{E}$ est décrite par une séquence de doublets (temps, déformation imposée) : $\mathcal{E} = ((t_i, \epsilon(t_i)), i = 1..N_\mathcal{E})$. La réponse expérimentale du matériau est donnée par $\mathcal{M}(\mathcal{E}) = (\sigma_{exp}(t_i), i = 1..N_\mathcal{E})$.

Notons $\mathcal{M}_R$ la loi de comportement représentée par le modèle rhéologique R, avec $\mathcal{M}_R(\mathcal{E}) = (\sigma_R(t, \mathcal{E}))$. On cherche ainsi à minimiser l'erreur de modélisation de $\mathcal{M}_R$ commise pour les différentes expériences $\mathcal{E}_1$, .. $\mathcal{E}_K$ disponibles :

$$\mathcal{F}(R) = \sum_{k=1}^{K} d(\mathcal{M}_R(\mathcal{E}_k),\ \mathcal{M}(\mathcal{E}_k))$$

L'erreur commise pour une expérience $\mathcal{E}$ est fixée dans ce qui suit à

$$d(\mathcal{M}_R(\mathcal{E}), \mathcal{M}(\mathcal{E})) = \sum_{i=1}^{N_\mathcal{E}} |\sigma_{exp}(t_i) - \sigma_R(t_i, \mathcal{E})|^2$$

3.3 Problème inverse — Problème direct

Pour que *tout* modèle rhéologique puisse être évalué (condition sine qua non pour pouvoir explorer tout l'espace), il faut donc pouvoir déterminer et simuler la loi de comportement correspondante. Le problème de déterminer "le" modèle collant aux données expérimentales exige aainsi de savoir simuler un modèle quelconque : on passe de la résolution d'un problème inverse à la résolution du problème direct.

Ce qui complique ici le problème direct, est qu'il doit être résolu en toute généralité : il faut réaliser un simulateur de modèles rhéologiques, opérationnel pour toutes architectures de modèle et tous coefficients numériques. Une possibilité consiste à déterminer analytiquement la loi de comportement correspondant au modèle rhéologique R donné, puis à calculer la réponse correspondant aux sollicitations imposées. Cette approche a été rejetée en raison de la complexité combinatoire due à la prise en compte d'un nombre *quelconque* de patins.

Nous avons donc préféré compiler et exécuter itérativement le modèle R donné, en fonction du trajet de chargement imposé. Le simulateur réalisé, appelé *X_RHEOL*, part de la description du modèle sous forme d'arbre ; à chaque noeud i de l'arbre sont associées des variables internes $\sigma_i(t)$ et $\epsilon_i(t)$, dont les relations dépendent de la nature du noeud et de l'état courant du modèle (voir ci-après). Le noeud racine de l'arbre donne l'état global du modèle en déformation et en contraintes.
A chaque pas de temps, *X_RHEOL* détermine les noeuds actifs de R (e.g. élagage des branches situées derrière un patin bloqué), pose le système d'équations liant les variables internes du modèle, auquel s'ajoute l'équation représentant la déformation imposée. La résolution de ce système donne l'état des contraintes internes du modèle, ainsi que sa réponse globale. Plus formellement, *X_RHEOL* comporte les étapes suivantes :

1. Initialiser à 0 l'état des noeuds internes du modèle;
2. Pour $i = 1, ..N_\mathcal{E}$
   - Déterminer les noeuds actifs du modèle compte tenu de l'état courant (*).

- Déterminer les équations de comportement élémentaires correspondant aux noeuds du modèle ;
- Ajouter l'équation décrivant la déformation imposée en t_i : $\epsilon_0(t_i) = \epsilon_{exp}(t_i)$
- Résoudre le système d'équations correspondant.

* La difficulté essentielle est due à la présence de patins : selon leur état, le comportement du modèle varie ainsi que les équations liant les variables d'état de nombreux noeuds : branches en parallèle avec un patin bloqué, noeuds en série situés derrière un patin,....

3.4 Résolution numérique

Soit j un noeud actif du modèle courant ; ce noeud est décrit à l'instant t_i par son état $\epsilon_j(t_i)$, et la tâche du simulateur est de déterminer la réponse $\sigma_j(t_i)$ correspondante. Selon le type du noeud j et l'état du modèle, les relations liant ϵ_j et σ_j sont bien connues [PER 60]. Nous avons remplacé les équations dérivées par des équations aux différences finies ; ceci permet pour chaque instant t_i de résoudre le système d'équations décrivant le comportement du modèle, comme un système d'équations linéaires.

- Cas d'un ressort de raideur k_j: $\sigma_j(t_i) = k_j \epsilon_j(t_i)$
- Cas d'un amortisseur de viscosité η_j $\sigma_j(t_i) = \eta_j \dot{\epsilon}_j(t_i) \approx \eta_j \frac{\epsilon_j(t_i)-\epsilon_j(t_{i-1})}{t_i - t_{i-1}}$
- Cas d'un patin de seuil σ_j : $|\sigma_j(t_i)| < \sigma_j$ OU ($(\dot{\epsilon}(t_i) = 0) \approx (\epsilon_j(t_i) = \epsilon_j(t_{i-1}))$)
- Cas des noeuds série et parallèle, où $j_1, ..j_J$ désignent les noeuds fils du noeud j :

cas Série	cas Parallèle
$\epsilon_j(t_i) = \epsilon_{j_1}(t_i) + .. + \epsilon_{j_J}(t_i)$	$\epsilon_j(t_i) = \epsilon_{j_1}(t_i) = .. = \epsilon_{j_J}(t_i)$
$\sigma_j(t_i) = \sigma_{j_1}(t_i) = .. = \sigma_{j_J}(t_i)$	$\sigma_j(t_i) = \sigma_{j_1}(t_i) + .. + \sigma_{j_J}(t_i)$

Le calcul de l'erreur de modélisation comporte ainsi lui-même trois sources d'erreur numérique : tout d'abord, l'erreur n'est mesurée que pour les instants $t_1, ..t_{N_\mathcal{E}}$ de l'expérience $\mathcal{E}$; ensuite, les équations dérivées liant ϵ et σ sont approchées par des différences finies ; enfin, la résolution du système linéaire comporte un bruit numérique intrinsèque.

Notons que l'optimisation par évolution est relativement robuste face à ce type d'erreurs : l'important est de savoir déterminer si un modèle est ou non meilleur qu'un autre, plus que de mesurer l'erreur exacte commise.
L'erreur intrinsèque pour un modèle R est mesurée comme la différence entre la réponse obtenue pour l'expérience $\mathcal{E}$ et la réponse obtenue pour $\mathcal{E}_{1/2}$, construite en retenant un point de $\mathcal{E}$ sur deux. Un modèle est considéré comme un succès si l'erreur de modélisation est inférieure ou égale à l'erreur intrinsèque.

4. Validation numérique

Cette approche a été validée par expérimentations numériques, et jusqu'à présent sur des données simulées. Pour plus de détails, voir [SCH 96]. Les résultats présentés concernent l'identification d'un modèle rhéologique connu (Fig. 1) avec les coefficients numériques suivants : $k_1 = 790.45$, $k_2 = 150.20$, $k_3 = 41.60$, $\eta = 6248.60$, $\sigma_S = 7.25$.

Les paramètres d'optimisation sont les suivants : 200 individus par populations, 200 générations, taux de croisement .4, taux de mutation .2, sélection par tournoi de taille 3. Le programme est écrit en C^{++} ; une exécution prend en moyenne 12 heures CPU sur Pentium 166 pour 150 points d'expériences. Les résultats présentés sont moyennés sur 15 exécutions indépendantes. Un taux de succès de 20% est obtenu ; i.e., 3 exécutions conduisent à au moins un modèle dont l'erreur de modélisation est voisine du bruit de calcul. Le meilleur individu déterminé est montré en Figure 2 (a), avec les coefficients $k_1 = 998.892$, $k_2 = 133.085$, $k_3 = 39.6647$, $\eta = 8698.78$, $\sigma_S = 19.0409$. Notons que cette structure est plus simple que le modèle cherché ; ceci est dû au fait que les trajets de chargement considérés ne comprennent pas de phase de recouvrance.

(a) Meilleur modèle identifié (b) Une erreur de modélisation

Figure 2: Résultats d'expérimentations numériques.

5. Conclusion

Ce travail est à notre connaissance le premier dans le cadre de l'identification automatique de lois rhéologiques (architecture et coefficients) à partir d'expériences. Il illustre l'intérêt des méthodes d'optimisation par évolution (algorithmes génétiques, programmation génétique), pour répondre à une large gamme de problèmes inverses mal posés.

De nombreuses extensions sont envisageables : la plus essentielle à court terme concerne la prise en compte de l'influence de la température (e.g. recherche des coefficients du modèle sous forme de fonctions linéaires de la température).

Bibliographie

[BOY 95] Boytard, K., Ouakka, A., and Van, K. D., (1995), "A modelling tool for the mechanical behavior of polyethylene pipe", dans *Pipeline Technology*(Denys, R., ed.), Elsevier.

[GOL 89] Goldberg, D. E., (1989), "Genetic algorithms in search, optimization and machine learning", Addison Wesley.

[KIC 96] Kichenin, J., Dang-Van, K., and Boytard, K., (1996), "Finite element simulation of new two dissipative mechanism models for bulk medium-density polyethylene", *J. of Mat. Sc.*, Vol 31, pp 1653–1661.

[KOZ 94] Koza, J. R., (1994), "Genetic programming II: Automatic discovery of reusable programs", Massachussetts: MIT Press.

[PER 60] Persoz, B., (1960), "Introduction à la rhéologie.", Dunod, Paris.

[SCH 96] Schoenauer, M., Sebag, M., Jouve, F., Lamy, B., and Maitournam, H., (1996), "23: Evolutionary identification of macro-mechanical models", dans *Advances in Genetic Programming II*(Angeline, P. J. and K. E. Kinnear, J., eds.), (Cambridge, MA), pp 467–488, MIT Press.

[SP 87] Sanchez-Palencia, E. and Zaoui, A., (1987), "Homogenization techniques for composite media", Lecture Notes in Physics, Springer Verlag.