

Implémentation

Interconversion

Prin. Corres.

Éléments finis

MEC6418 - NOTES DE COURS

Calcul de structure avec des matériaux viscoélastiques

Par: Martin Lévesque
professeur du département de génie mécanique

▷ Implémentation

Introduction

Diff. finies

Algorithme

Interconversion

Prin. Corres.

Éléments finis

Implémentation numérique

→ Les lois de comportement

$$\boldsymbol{\sigma}(t) = \mathbf{C}^{(0)} : \boldsymbol{\varepsilon}(t) + \int_0^t \sum \mathbf{C}^{(i)} \exp[-\omega_i(t - \tau)] : \frac{d\boldsymbol{\varepsilon}}{d\tau} d\tau \quad (1)$$

$$\boldsymbol{\varepsilon}(t) = \mathbf{S}^{(0)} : \boldsymbol{\sigma}(t) + \int_0^t \sum \mathbf{S}^{(i)} (1 - \exp[-\lambda_i(t - \tau)]) : \frac{d\boldsymbol{\sigma}}{d\tau} d\tau \quad (2)$$

- font intervenir l'histoire de contraintes et de déformations.
- Calculer $\boldsymbol{\sigma}$ ou $\boldsymbol{\varepsilon}$ à une valeur de t nécessite d'intégrer l'histoire de $\boldsymbol{\varepsilon}$ et $\boldsymbol{\sigma}$ de $\tau = 0$ à $\tau = t$.
- Pour des histoires arbitraires de $\boldsymbol{\varepsilon}$ et $\boldsymbol{\sigma}$, ce calcul doit se faire de manière numérique.
- Plusieurs stratégies existent dans la littérature pour accomplir cette tâche.
 - La majeure partie des méthodes se concentrent sur la discrétisation de l'intégrale et utilisent les propriétés des exponentielles pour obtenir une expression récursive.

Implémentation

▷ Introduction

Diff. finies

Algorithme

Interconversion

Prin. Corres.

Éléments finis

Implémentation

▷ Introduction

Diff. finies

Algorithme

Interconversion

Prin. Corres.

Éléments finis

- Dans la suite de ce document, nous allons présenter une approche récemment proposée par Crochon *et al.* dans la revue Mech. Time-Depend. Mater. (2010) 14:359 – 387.
- La méthode utilise le fait que la forme intégrale des lois viscoélastiques provient d'une équation différentielle.
- Cette forme est propice à l'utilisation des schémas de différence finie (Euler, Crank-Nicholson, Runge-Kutta) et est plus précise que toutes les autres méthodes existantes.

→ Supposons que l'on ait une équation différentielle du type:

$$y' = f(t, y) \quad \text{avec} \quad y(0) = y_0 \quad (3)$$

où $y = y(t)$ et t est une variable. Tout dépendant de la forme de f , on pourra trouver une solution analytique à cette équation différentielle.

- Quand on ne peut solutionner analytiquement, on peut trouver une solution approximative en utilisant un schéma de différences finies.
- L'approche consiste à discrétiser le domaine de t pour lequel on veut la solution en un nombre fini d'incrément. Par exemple, si chaque incrément a la même longueur h , on aura que le temps $t_n = nh$, où n est un entier.
- On notera par $y_n = y(t_n)$.

Implémentation

Introduction

▷ Diff. finies

Algorithme

Interconversion

Prin. Corres.

Éléments finis

→ De l'équation (3), on peut voir que $f(t, y)$ est la pente de y . Alors, une première approximation pourrait être que:

$$y_{n+1} \approx y_n + hf(t_n, y_n) \quad (4)$$

Cette approche est appelée schéma d'Euler avant et est explicite car la solution à t_{n+1} ne dépend que de y_n . Ce schéma est reconnu être instable et peu précis.

→ Une autre approximation est de calculer

$$y_{n+1} \approx y_n + hf(t_{n+1}, y_{n+1}) \quad (5)$$

où cette fois le schéma est implicite (la majeure partie du temps) et est appelé schéma d'Euler arrière. Ce schéma est stable mais nécessite plusieurs pas de temps pour une bonne précision (sa précision est de l'ordre de h).

Implémentation

Introduction

▷ Diff. finies

Algorithme

Interconversion

Prin. Corres.

Éléments finis

→ Une autre approche serait d'utiliser la pente moyenne sur l'incrément de temps de sorte que

$$y_{n+1} \approx y_n + \frac{h}{2} [f(t_n, y_n) + f(t_{n+1}, y_{n+1})] \quad (6)$$

Cette approche est appelée schéma de Crank-Nicholson et est implicite. De plus, sa précision est de l'ordre de h^2 , ce qui lui permet de converger plus rapidement que le schéma d'Euler.

→ On peut définir d'autres schémas en utilisant des développements de Taylor de manière à faire disparaître certains termes. Par exemple, la méthode de Runge-Kutta explicite permet de calculer

$$y_{n+1} \approx y_n + \frac{h}{6} [k_1 + 2k_2 + 2k_3 + k_4] \quad (7)$$

où $k_1 = f(t_n, y_n)$, $k_2 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1)$,
 $k_3 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2)$ et $k_4 = f(t_{n+1}, y_n + hk_3)$

Implémentation

Introduction

▷ Diff. finies

Algorithme

Interconversion

Prin. Corres.

Éléments finis

→ Dans le chapitre sur le développement de la loi de comportement viscoélastique, on avait obtenu que:

$$\begin{aligned}\boldsymbol{\sigma}(t) &= \mathbf{L}_1 : \boldsymbol{\varepsilon}(t) + \mathbf{L}_2 : \boldsymbol{\xi}(t) \\ \mathbf{B} : \dot{\boldsymbol{\xi}} + \mathbf{L}_3 : \boldsymbol{\xi} + \mathbf{L}_2^\top : \boldsymbol{\varepsilon} &= \mathbf{0}\end{aligned}\quad (8)$$

→ Si on reprend la notation générale de l'équation (3), on a ici que

$$\dot{\boldsymbol{\xi}} = -\mathbf{B}^{-1} : \left(\mathbf{L}_3 : \boldsymbol{\xi} + \mathbf{L}_2^\top : \boldsymbol{\varepsilon} \right) \quad (9)$$

→ Si on utilise un schéma d'Euler implicite, on aura que:

$$\begin{aligned}\boldsymbol{\xi}^{n+1} &= \boldsymbol{\xi}^n + h\dot{\boldsymbol{\xi}}^{n+1} \\ &= \boldsymbol{\xi}^n - h\mathbf{B}^{-1} : \left(\mathbf{L}_3 : \boldsymbol{\xi}^{n+1} + \mathbf{L}_2^\top : \boldsymbol{\varepsilon}^{n+1} \right)\end{aligned}\quad (10)$$

d'où on tire que:

$$\boldsymbol{\xi}^{n+1} = \left(\mathbf{I} + h\mathbf{B}^{-1} : \mathbf{L}_3 \right)^{-1} : \left(\boldsymbol{\xi}^n - h\mathbf{B}^{-1} : \mathbf{L}_2^\top : \boldsymbol{\varepsilon}^{n+1} \right) \quad (11)$$

Implémentation

Introduction

Diff. finies

▷ Algorithmme

Interconversion

Prin. Corres.

Éléments finis

→ L'équation (11) peut se mettre sous la forme:

$$\xi^{n+1} = \mathbf{W}_1 : \xi^n + \mathbf{W}_2 : \varepsilon^{n+1} \quad (12)$$

→ Avec la loi de comportement, on a que:

$$\begin{aligned} \sigma^{n+1} &= \mathbf{L}_1 : \varepsilon^{n+1} + \mathbf{L}_2 : \xi^{n+1} \\ &= \mathbf{L}_1 : \varepsilon^{n+1} + \mathbf{L}_2 : (\mathbf{W}_1 : \xi^n + \mathbf{W}_2 : \varepsilon^{n+1}) \\ &= (\mathbf{L}_1 + \mathbf{L}_2 : \mathbf{W}_2) : \varepsilon^{n+1} + \mathbf{L}_2 : \mathbf{W}_1 : \xi^n \end{aligned} \quad (13)$$

→ On voit donc que σ^{n+1} dépend de ε^{n+1} et de ξ^n .

→ L'algorithme va consister dans un premier temps à discrétiser l'histoire de chargement (i.e. $\varepsilon(t)$) en N incréments de temps. Par la suite, pour chaque pas de temps, on calcule σ^{n+1} avec l'équation (13) et ξ^{n+1} avec l'équation (11). On pourra ainsi calculer la réponse σ pour les N pas de temps choisis.

Implémentation

Introduction

Diff. finies

▷ Algorithmme

Interconversion

Prin. Corres.

Éléments finis

Implémentation

Introduction

Diff. finies

▷ Algorithmme

Interconversion

Prin. Corres.

Éléments finis

- Le problème avec l'algorithme précédent est qu'il fait intervenir les matrices internes \mathbf{B} et \mathbf{L}_i alors que la loi de comportement fera intervenir les $\mathbf{C}^{(i)}$ et ω_i . Il y a un lien entre ces quantités et l'objectif des prochains transparents est de l'explicitier.
- Dans les transparents sur la viscoélasticité, on avait obtenu que

$$\sigma_i(t) = \left(L_{1ij} - \frac{L_{2ir} L_{2jr}}{L_{3rr}} \right) \varepsilon_j(t) + \int_0^t \frac{L_{2ir} L_{2jr}}{L_{3rr}} \exp \left[-\frac{L_{3rr}}{B_{rr}} (t - \tau) \right] \frac{d\varepsilon_j(t)}{d\tau} d\tau \quad (14)$$

où toutes les matrices sont dans la base qui rendent \mathbf{B} et \mathbf{L}_3 diagonales.

- Lorsque deux matrices sont symétriques et définies positives elles sont simultanément diagonalisables. De plus, une des deux matrices diagonalisées sera la matrice identité. Dans notre cas, supposons que la matrice $\mathbf{B} = \mathbf{I}$ lorsque diagonalisée. Avec l'équation (1), on peut voir que les éléments de la matrice \mathbf{L}_3 diagonalisée doivent être égaux au ω_i .

→ Avec cette observation, on peut voir que:

$$\frac{1}{\omega_i} \mathbf{L}_2^{(i)} : \left(\mathbf{L}_2^{(i)} \right)^T = \mathbf{C}^{(i)} \quad (15)$$

où $\mathbf{L}_2^{(i)}$ est une matrice qui fait partie de \mathbf{L}_2 . On peut ré-écrire cette équation sous la forme:

$$\mathbf{L}_2^{(i)} : \left(\mathbf{L}_2^{(i)} \right)^T = \omega_i \mathbf{C}^{(i)} \quad (16)$$

→ Comme $\mathbf{C}^{(i)} \geq 0$ et symétrique, et que $\omega_i > 0$, $\omega_i \mathbf{C}^{(i)}$ permet une décomposition de Cholesky.

→ Par définition, une décomposition de Choleski permet d'exprimer une matrice symétrique sous la forme:

$$\mathbf{A} = \mathbf{L} : \mathbf{L}^T \quad (17)$$

où la matrice \mathbf{L} est une matrice triangulaire inférieure.

Implémentation

Introduction

Diff. finies

▷ Algorithmme

Interconversion

Prin. Corres.

Éléments finis

Implémentation

Introduction

Diff. finies

▷ Algorithme

Interconversion

Prin. Corres.

Éléments finis

- Intéressons nous aux dimensions des diverses matrices. La matrice \mathbf{L}_2 servira à représenter les différents $\mathbf{C}^{(i)}$. Si on a N ω_i , et que chaque matrice $\mathbf{C}^{(i)}$ comprend 36 termes (parce que c'est une matrice 6×6), la matrice \mathbf{L}_2 doit avoir $36N$ termes.
- La matrice \mathbf{L}_2 est une matrice $L_{2_{i_r}}$ où $i \in [1, 6]$ car $\mathbf{L}_2 : \xi$ doit avoir les mêmes dimensions que les contraintes, c'est à dire 6 composantes. L'indice r correspond donc au nombre de termes dans ξ .
- Comme $i \times r = 6r = 36N$, il faut que $r = 6N$.
 - Donc, si on a 2 ω_i , \mathbf{L}_2 est une matrice 6×12 .
 - Alors, la matrice \mathbf{L}_2 pourra être exprimée comme:

$$\mathbf{L}_2 = \left(\mathbf{L}_2^{(1)} \mid \mathbf{L}_2^{(2)} \mid \dots \mid \mathbf{L}_2^N \right) \quad (18)$$

où chaque $\mathbf{L}_2^{(i)}$ est une matrice 6×6 triangulaire inférieure qui est la décomposition de Cholesky de $\omega_i \mathbf{C}^{(i)}$.

Implémentation

Introduction

Diff. finies

▷ Algorithmme

Interconversion

Prin. Corres.

Éléments finis

→ Comme $r = 6N$, \mathbf{B} sera la matrice identité $6N \times 6N$ et \mathbf{L}_3 sera une matrice diagonale $6N \times 6N$ où les termes diagonaux seront les ω_i . On aura par exemple, $L_{3_{rr}} = \omega_1$ pour $r \in [1, 6]$, $L_{3_{rr}} = \omega_2$ pour $r \in [7, 12]$, etc.

→ On a que $\mathbf{C}^{(0)} = L_{1_{ij}} - \frac{L_{2_{ir}} L_{2_{jr}}}{L_{3_{rr}}} = \mathbf{L}_1 - \sum_{i=1}^{i=N} \mathbf{C}^{(i)}$. On aura donc que:

$$\mathbf{L}_1 = \sum_{i=0}^{i=N} \mathbf{C}^{(i)} \quad (19)$$

→ On a donc tout ce qu'il faut pour programmer une implémentation numérique.

(Voir exemple Mathematica sur le site web)

Implémentation

▷ Interconversion

Introduction

Algorithme

Prin. Corres.

Éléments finis

Problème d'interconversion

Implémentation

Interconversion

▷ Introduction

Algorithme

Prin. Corres.

Éléments finis

→ Le problème d'interconversion consiste à calculer la souplesse de fluage à partir du module de relaxation, ou vice-versa.

→ Ce n'est pas un problème simple car:

$$\mathbf{S}(t) \neq (\mathbf{C}(t))^{-1} \quad (20)$$

→ Les transformées de Laplace-Carson et le principe de correspondance permettent d'établir une solution analytique à ce problème, conceptuellement.

- La solution numérique l'équation analytique pose de gros problèmes.
- Pendant plusieurs années, depuis les années 60, les chercheurs ont eu recours à plusieurs artifices pour contourner le problème.

→ Cyr, Crochon, Li et Lévesque (Mechanics of Time Dependent Materials, v. 16, pp. 53 – 82, 2013) ont proposé une solution stable au problème et c'est cet article qui est présenté dans les prochains transparents.

→ A partir de la thermodynamique, on a obtenu que:

$$\begin{aligned}\boldsymbol{\sigma}(t) &= \mathbf{L}_1 : \boldsymbol{\varepsilon}(t) + \mathbf{L}_2 : \boldsymbol{\xi}(t) \\ \mathbf{B} : \dot{\boldsymbol{\xi}} + \mathbf{L}_3 : \boldsymbol{\xi} + \mathbf{L}_2^\top : \boldsymbol{\varepsilon} &= \mathbf{0}\end{aligned}\tag{21}$$

→ On peut exprimer $\boldsymbol{\varepsilon}$ comme:

$$\begin{aligned}\boldsymbol{\varepsilon} &= \mathbf{L}_1^{-1} : [\boldsymbol{\sigma} - \mathbf{L}_2 : \boldsymbol{\xi}] \\ &= \mathbf{A}_1 : \boldsymbol{\sigma} - \mathbf{A}_2 : \boldsymbol{\xi} \\ &= \mathbf{A}_1 : \boldsymbol{\sigma} - \mathbf{A}_2 : \boldsymbol{\chi}\end{aligned}\tag{22}$$

où les $\boldsymbol{\xi}$ ont été renommés à $\boldsymbol{\chi}$ pour être consistant à la présentation faite dans les autres chapitres.

Implémentation

Interconversion

Introduction

▷ Algorithmme

Prin. Corres.

Éléments finis

→ Injecter l'équation (22) dans l'équation (21) conduit à:

$$\begin{aligned}
 \mathbf{0} &= \mathbf{B} : \dot{\boldsymbol{\xi}} + \mathbf{L}_3 : \boldsymbol{\xi} + \mathbf{L}_2^\top : \mathbf{L}_1^{-1} : [\boldsymbol{\sigma} - \mathbf{L}_2 : \boldsymbol{\xi}] \\
 &= \mathbf{B} : \dot{\boldsymbol{\xi}} + \left[\mathbf{L}_3 - \mathbf{L}_2^\top : \mathbf{L}_1^{-1} : \mathbf{L}_2 \right] : \boldsymbol{\xi} + \mathbf{L}_2^\top : \mathbf{L}_1^{-1} : \boldsymbol{\sigma} \\
 &= \mathbf{B} : \dot{\boldsymbol{\chi}} + \mathbf{A}_3 : \boldsymbol{\chi} + \mathbf{A}_2^\top : \boldsymbol{\sigma}
 \end{aligned}
 \tag{23}$$

→ Comme on a vu dans le chapitre sur la viscoélasticité, la loi de comportement est:

$$\begin{aligned}
 \boldsymbol{\varepsilon}(t) &= \mathbf{A}_1 : \boldsymbol{\sigma}(t) - \mathbf{A}_2 : \boldsymbol{\chi}(t) \\
 \mathbf{B} : \dot{\boldsymbol{\chi}} + \mathbf{A}_3 : \boldsymbol{\chi} + \mathbf{A}_2^\top : \boldsymbol{\sigma} &= \mathbf{0}
 \end{aligned}
 \tag{24}$$

Implémentation

Interconversion

Introduction

▷ Algorithmme

Prin. Corres.

Éléments finis

→ Le lien explicite entre les \mathbf{A}_i et \mathbf{L}_i est le suivant:

$$\mathbf{A}_1 = \mathbf{L}_1^{-1} \quad (25a)$$

$$\mathbf{A}_2 = \mathbf{L}_1^{-1} : \mathbf{L}_2 \quad (25b)$$

$$\mathbf{A}_3 = \mathbf{L}_3 - \mathbf{L}_2^T : \mathbf{L}_1^{-1} : \mathbf{L}_2 \quad (25c)$$

→ On a vu sur le chapitre sur la viscoélasticité que $\mathbf{B} \geq 0$ et $\mathbf{A}_3 \geq 0$. Alors, si on appelle \mathbf{P} la matrice qui contient les vecteurs propres de \mathbf{A}_3 de telle sorte que:

$$\mathbf{A}_3 = \mathbf{P} : \mathbf{D} : \mathbf{P}^T \quad (26)$$

où \mathbf{D} est une matrice diagonale qui contient les valeurs propres de \mathbf{A}_3 et que l'on pose que $\chi = \mathbf{P} : \chi^*$, on aura que:

$$\begin{aligned} \mathbf{0} &= \mathbf{P}^T : \mathbf{B} : \mathbf{P} : \dot{\chi}^* + \mathbf{P}^T : \mathbf{A}_3 : \mathbf{P} : \chi^* \\ &\quad + \mathbf{P}^T : \mathbf{A}_2^T : \sigma \\ &= \mathbf{B}^* : \dot{\chi}^* + \mathbf{A}_3^* : \dot{\chi}^* + \mathbf{A}_2^{*T} : \sigma \end{aligned} \quad (27)$$

Implémentation

Interconversion

Introduction

▷ Algorithmme

Prin. Corres.

Éléments finis

→ Avec le changement de variables $\chi = \mathbf{P} : \chi^*$, la loi de comportement devient:

$$\begin{aligned} \varepsilon(t) &= \mathbf{A}_1 : \boldsymbol{\sigma}(t) - \mathbf{A}_2 : \mathbf{P} : \chi^*(t) = \mathbf{A}_1 : \boldsymbol{\sigma} - \mathbf{A}_2^* : \chi^* \\ \mathbf{B}^* : \dot{\chi}^* + \mathbf{A}_3^* : \dot{\chi}^* + \mathbf{A}_2^{*\top} : \boldsymbol{\sigma} &= \mathbf{0} \end{aligned} \quad (28)$$

→ La solution de l'équation différentielle de l'équation (28) conduit à:

$$\begin{aligned} \varepsilon_i(t) &= A_{1_{ij}} \sigma_j + \frac{A_{2_{ir}}^* A_{2_{jr}}^*}{A_{3_{rr}}^*} \int_0^t \left(1 - \exp \left[-\frac{A_{3_{rr}}^*}{B_{rr}^*} (t - \tau) \right] \right) \\ &\quad \times \frac{d\sigma_j}{d\tau} d\tau \end{aligned} \quad (29)$$

→ Cela conduit à la loi de comportement:

$$\begin{aligned} \varepsilon(t) &= \mathbf{S}^{(0)} : \boldsymbol{\sigma}(t) \\ &\quad + \sum_i \int_0^t \mathbf{S}^{(i)} (1 - \exp[-\lambda_i(t - \tau)]) : \frac{d\boldsymbol{\sigma}}{d\tau} d\tau \end{aligned} \quad (30)$$

Implémentation

Interconversion

Introduction

▷ Algorithmme

Prin. Corres.

Éléments finis

Implémentation

Interconversion

Introduction

▷ Algorithmme

Prin. Corres.

Éléments finis

→ L'algorithme d'interconversion consiste donc à:

1. Exprimer les matrices \mathbf{L}_i à partir des matrices $\mathbf{C}^{(i)}$ à l'aide des décompositions de Choleski et les hypothèses vues pour l'implémentation numérique.
2. Exprimer les matrices \mathbf{A}_i à partir des matrices \mathbf{L}_i selon l'équation (25).
3. Calculer la matrice \mathbf{P} qui contient en colonnes les vecteurs propres orthonormés de \mathbf{A}_3 . Cela peut se faire de manière stable et très précis avec l'algorithme *Single Value Decomposition*.
4. Calculer les matrices \mathbf{B}^* ($= \mathbf{I}$), \mathbf{A}_3^* et \mathbf{A}_2^* .
5. Calculer les $\mathbf{S}^{(i)}$ à partir des matrices \mathbf{B}^* , \mathbf{A}_3^* et \mathbf{A}_2^* .

NOTE

→ Nous venons de démontrer l'interconversion de $\mathbf{C}(t) \rightarrow \mathbf{S}(t)$. Tout comme pour l'implémentation numérique des lois viscoélastiques, on peut faire l'interconversion $\mathbf{S}(t) \rightarrow \mathbf{C}(t)$ avec une approche similaire contenant de subtiles différences (**voir exercices proposés**).

Implémentation

Interconversion

▷ Prin. Corres.

Introduction

Princ. Corres.

Conv. Prop.

Laplace-Carson

Inversion

Éléments finis

Principe de correspondance

Implémentation

Interconversion

Prin. Corres.

▷ Introduction

Princ. Corres.

Conv. Prop.

Laplace-Carson

Inversion

Éléments finis

- Le calcul de structure avec des matériaux viscoélastiques linéaires est plus compliqué qu'avec des matériaux élastiques linéaires car ils introduisent une dépendance au temps.
- On l'a vu dans le développement de la loi de comportement, un outil qui permet de faire disparaître la dépendance au temps est la transformée de Laplace.
- Par définition, la transformée de Laplace de la fonction f est:

$$\mathcal{L}(f) = \tilde{f}(s) = \int_0^{\infty} \exp[-st] f(t) dt \quad (31)$$

- On peut donc remarquer qu'elle ne s'applique que sur l'aspect temporel du problème. L'aspect spatial n'est aucunement affecté.
- La transformée de Laplace permet donc de découpler la dépendance temps-espace associée aux problèmes viscoélastiques.
- La démarche sera d'appliquer la transformée de Laplace sur le problème, le solutionner dans l'espace de Laplace et appliquer la transformée inverse pour le retrouver dans l'espace réel.

Implémentation

Interconversion

Prin. Corres.

Introduction

▷ Princ. Corres.

Conv. Prop.

Laplace-Carson

Inversion

Éléments finis

- L'outil qui nous permet de solutionner ces problèmes se nomme principe de correspondance viscoélastique linéaire.
- Pour qu'il puisse s'appliquer, le problème doit avoir les particularités suivantes:
 1. Tous les matériaux doivent être linéaires.
 2. Tous les paramètres définissant les matériaux ne doivent pas dépendre du temps.
 3. En chaque point du problème, le type de conditions aux rives doit rester le même. Par exemple, on ne pourrait pas avoir une force imposée durant un certain temps et par la suite un déplacement imposé au même point.
 4. Il ne faut pas que la géométrie change avec le temps.

→ Considérons la loi de comportement:

$$\boldsymbol{\varepsilon}(t) = \int_0^t \mathbf{S}(t - \tau) : \frac{d\boldsymbol{\sigma}}{d\tau} d\tau \quad (32)$$

→ Si l'on applique la transformée de Laplace à cette équation, on obtiendra:

$$\tilde{\boldsymbol{\varepsilon}} = s\tilde{\mathbf{S}} : \tilde{\boldsymbol{\sigma}} \quad (33)$$

→ Considérons la loi de comportement:

$$\boldsymbol{\sigma}(t) = \int_0^t \mathbf{C}(t - \tau) : \frac{d\boldsymbol{\varepsilon}}{d\tau} d\tau \quad (34)$$

→ Si l'on applique la transformée de Laplace à cette équation, on obtiendra:

$$\tilde{\boldsymbol{\sigma}} = s\tilde{\mathbf{C}} : \tilde{\boldsymbol{\varepsilon}} \quad (35)$$

Implémentation

Interconversion

Prin. Corres.

Introduction

Princ. Corres.

▷ Conv. Prop.

Laplace-Carson

Inversion

Éléments finis

→ Des équations (33) et (35) on tire que:

$$\tilde{\sigma} = \frac{1}{s} \tilde{\mathbf{S}}^{-1} : \tilde{\epsilon} = s \tilde{\mathbf{C}} : \tilde{\epsilon} \quad (36)$$

→ On aura donc que:

$$\tilde{\mathbf{S}} = \frac{1}{s^2} \tilde{\mathbf{C}}^{-1} \quad (37)$$

→ D'où l'on tirera que:

$$\mathbf{S}(t) = \mathcal{L}^{-1} \left(\frac{1}{s^2} \tilde{\mathbf{C}}^{-1} \right) \quad (38)$$

→ Cette relation est très utile car elle permet de calculer la souplesse de fluage si le module de relaxation est connu, ou vice versa.

Implémentation

Interconversion

Prin. Corres.

Introduction

Princ. Corres.

▷ Conv. Prop.

Laplace-Carson

Inversion

Éléments finis

Implémentation

Interconversion

Prin. Corres.

Introduction

Princ. Corres.

Conv. Prop.

▷ Laplace-Carson

Inversion

Éléments finis

→ Un outil qui est plus “ physique ” que la transformée de Laplace est la transformée de Laplace – Carson. La transformée de Laplace-Carson f^* d'une fonction $f(t)$ est définie par:

$$f^*(s) = s\mathcal{L}(f) \quad (39)$$

→ Alors, si l'on reprend les deux lois de comportement auxquelles on a appliqué la transformée de Laplace et que l'on multiplie par s de chaque côté de l'équation, on a:

$$\varepsilon^* = \mathbf{S}^* : \sigma^* \quad (40a)$$

$$\sigma^* = \mathbf{C}^* : \varepsilon^* \quad (40b)$$

→ Ce qui permet d'obtenir:

$$\mathbf{S}^* = (\mathbf{C}^*)^{-1} \quad (41)$$

Implémentation

Interconversion

Prin. Corres.

Introduction

Princ. Corres.

Conv. Prop.

▷ Laplace-Carson

Inversion

Éléments finis

$$\mathbf{S}^* = (\mathbf{C}^*)^{-1}$$

- Ce résultat est très similaire à celui que l'on obtiendrait en élasticité linéaire.
- Alors, en pratique, on utilisera la transformée de Laplace – Carson pour solutionner les problèmes.
- Si pour un problème élastique on connaît la solution en fonction des propriétés mécaniques du matériau, la démarche est très simple:
 - On applique la transformée de Laplace – Carson à toutes les propriétés mécaniques.
 - Dans la solution du problème élastique, on remplace les propriétés élastiques par les propriétés transformées.
 - On obtient la solution réelle en appliquant la transformée inverse.

(Voir exemples d'application faits au tableau)

Implémentation

Interconversion

Prin. Corres.

Introduction

Princ. Corres.

Conv. Prop.

Laplace-Carson

▷ Inversion

Éléments finis

- On vient de voir que l'obtention de la solution temporelle peut rapidement devenir compliquée.
- De plus, dans la majorité des cas, il n'existe pas de solution analytique pour la réponse temporelle.
- On doit donc avoir recours à des méthodes numériques, comme la méthode de Collocation introduite par Schapery en 1962.
- Nommons par $\psi(t)$ la solution temporelle à un problème viscoélastique solutionné par le principe de correspondance. Alors, on peut supposer que $\psi(t)$ peut se décomposer comme:

$$\psi(t) = \psi' + \psi''t + \Delta\psi(t) \quad (42)$$

où ψ' et ψ'' sont scalaires.

- En général, $\psi(t)$ va être bornée. En effet, pour une déformation ou contrainte appliquée finie, la réponse est aussi finie, sauf pour des cas singuliers. Pour ce cours, on supposera que

$$\psi(t) = \psi' + \Delta\psi(t) \quad (43)$$

Implémentation

Interconversion

Prin. Corres.

Introduction

Princ. Corres.

Conv. Prop.

Laplace-Carson

▷ Inversion

Éléments finis

→ Pour arriver à une approximation de $\psi(t)$, Schapery a proposé $\Delta\psi_d(t)$ sous la forme d'une série de Dirichlet pour approximer $\Delta\psi(t)$ de sorte que:

$$\Delta\psi(t) \approx \Delta\psi_d(t) = \sum_{i=1}^N \phi_i \exp[-t\tau_i] \quad (44)$$

où les $\tau_i > 0$ et choisis à priori.

→ On peut voir que $\Delta\psi_s(\infty) = 0$.

→ Une des propriétés des transformées de Laplace – Carson est le théorème de la valeur finale. Supposons que l'on veuille calculer la limite suivante en utilisant les propriétés de dérivation des transformées de Laplace:

$$\lim_{s \rightarrow 0} \int_0^{\infty} \dot{f}(t) \exp[-st] dt = \lim_{s \rightarrow 0} [s\mathcal{L}(f) - f(0)] \quad (45)$$

Implémentation

Interconversion

Prin. Corres.

Introduction

Princ. Corres.

Conv. Prop.

Laplace-Carson

▷ Inversion

Éléments finis

→ Comme l'intégrale se fait par rapport à t , on peut entrer la limite dans l'intégrale, ce qui nous conduira à:

$$\int_0^{\infty} \dot{f}(t) dt = \lim_{s \rightarrow 0} [f^*(s) - f(0)] \quad (46)$$

et après intégration on aura:

$$f(\infty) - f(0) = f^*(0) - f(0) \quad (47)$$

ce qui conduit au théorème de la valeur limite:

$$f(\infty) = f^*(0) \quad (48)$$

→ En appliquant ce résultat à $\psi(t)$ en utilisant la définition de $\Delta\psi(t)$, on aura que:

$$\psi' = \psi^*(0) \quad (49)$$

ce qui nous permettra de définir:

$$\Delta\psi(t) = \psi(t) - \psi^*(0) \quad (50)$$

Implémentation

Interconversion

Prin. Corres.

Introduction

Princ. Corres.

Conv. Prop.

Laplace-Carson

▷ Inversion

Éléments finis

→ La méthode de collocation de Schapery consiste à trouver $\Delta\psi_d(t)$ telle que la quantité suivante soit minimisée:

$$E = \int_0^{\infty} [\Delta\psi(t) - \Delta\psi_d(t)]^2 dt \quad (51)$$

→ Dans cette équation, les seules variables sont les ϕ_i . Alors, E sera minimale lorsque:

$$\frac{\partial E}{\partial \phi_i} = -2 \int_0^{\infty} [\Delta\psi(t) - \Delta\psi_d(t)] \exp[-t\tau_i] dt = 0 \quad (52)$$

→ Si on multiplie chaque côté de l'équation par τ_i , on a que:

$$\tau_i \int_0^{\infty} [\Delta\psi(t) - \Delta\psi_d(t)] \exp[-t\tau_i] dt = 0 \quad (53)$$

ce qui est en fait la définition de la transformée de Laplace – Carson (ici, $s = \tau_i$), ce qui permettra d'écrire:

$$\Delta\psi^*(\tau_i) = \Delta\psi_d^*(\tau_i) \quad \text{pour } i \in [1, N] \quad (54)$$

Implémentation

Interconversion

Prin. Corres.

Introduction

Princ. Corres.

Conv. Prop.

Laplace-Carson

▷ Inversion

Éléments finis

$$\Delta\psi^*(\tau_i) = \Delta\psi_d^*(\tau_i) \quad \text{pour } i \in [1, N]$$

→ Cette dernière équation est en fait un système à N équations où les inconnues sont les ϕ_i .

→ On aura que:

$$\begin{aligned} \Delta\psi_d^*(s) &= s\mathcal{L} \left(\sum_{j=1}^N \phi_j \exp[-t\tau_j] \right) \\ &= \sum_{j=1}^N \phi_j \frac{s}{s + \tau_j} \end{aligned} \quad (55)$$

→ Ainsi:

$$\begin{aligned} \Delta\psi_d^*(\tau_i) &= \sum_{j=1}^N \phi_j \frac{\tau_i}{\tau_i + \tau_j} \\ &= \zeta_{ij} \phi_j \end{aligned} \quad (56)$$

→ Ceci conduira au système linéaire d'équations:

$$\Delta\psi^*(\tau_i) = \zeta_{ij}\phi_j$$

avec

$$\zeta_{ij} = \frac{\tau_i}{\tau_i + \tau_j} \quad \text{et} \quad \Delta\psi^*(\tau_i) = \psi^*(\tau_i) - \psi^*(0)$$

→ La solution de ce système sera:

$$\phi_i = (\zeta_{ij})^{-1} \Delta\psi^*(\tau_j) \quad (57)$$

→ La solution approchée

$$\psi_d(t) = \psi' + \Delta\psi_d(t) = \psi^*(0) + \sum_{i=1}^N \phi_i \exp[-t\tau_i] \quad (58)$$

dépendra donc des τ_i choisis ainsi que de leur nombre.

Implémentation

Interconversion

Prin. Corres.

Introduction

Princ. Corres.

Conv. Prop.

Laplace-Carson

▷ Inversion

Éléments finis

Implémentation

Interconversion

Prin. Corres.

Introduction

Princ. Corres.

Conv. Prop.

Laplace-Carson

▷ Inversion

Éléments finis

→ En principe, plus on prendra de termes dans la série de Dirichlet, plus l'approximation sera précise. Toutefois, comme pour les séries de Taylor, lorsque le nombre de termes est grand on peut obtenir des oscillations importantes. Il est donc nécessaire de choisir astucieusement le nombre de termes dans la série.

Implémentation

Interconversion

Prin. Corres.

▷ Éléments finis

Principe

Imp. générale

Utilisation

Utilisation code de calculs par éléments finis

Implémentation

Interconversion

Prin. Corres.

Éléments finis

▷ Principe

Imp. générale

Utilisation

- Lorsqu'un problème d'éléments finis présente une nonlinéarité (par exemple géométrique ou matérielle), la solution est faite de manière incrémentale.
- La Figure 1 illustre schématiquement le processus (où les contraintes devraient être remplacées par des forces et les déformations par des déplacements)
- Par exemple, si on se trouve au temps t_n , on aura σ^n et ε^n . Si ε augmente de $\Delta\varepsilon$, alors, $\sigma^{n+1} = \sigma^n + \Delta\sigma$, où $\Delta\sigma = f(\Delta\varepsilon)$ et où f est non linéaire.

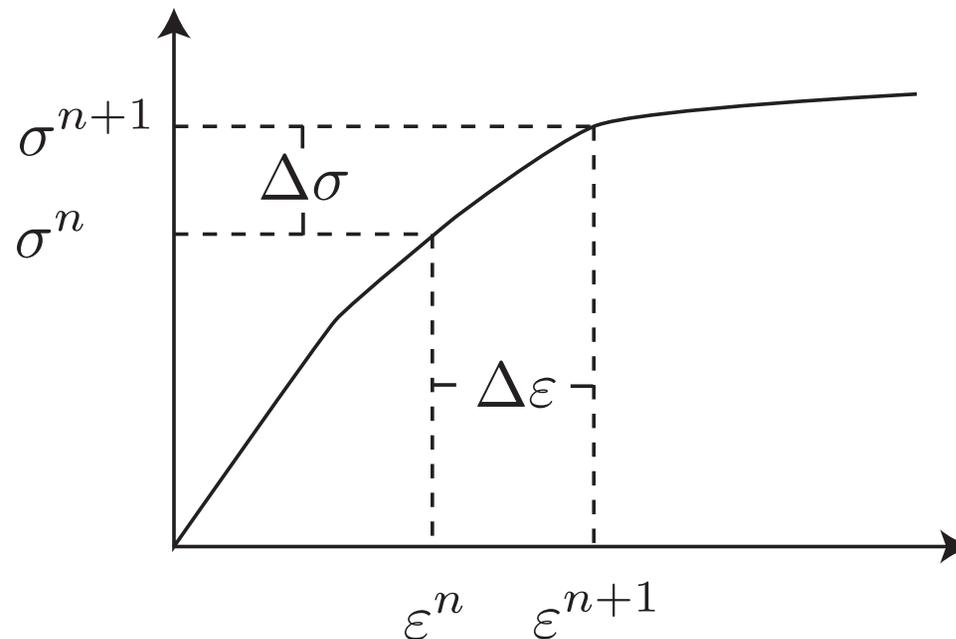


Figure 1: Schématisation d'un schéma incrémental de solution

Implémentation

Interconversion

Prin. Corres.

Éléments finis

▷ Principe

Imp. générale

Utilisation

- Supposons que l'on connaisse $\Delta\sigma$ et que l'on veuille calculer $\Delta\varepsilon$. Une manière de solutionner l'équation $\Delta\sigma = f(\Delta\varepsilon)$ est d'utiliser la méthode itérative de Newton.
- Ré-écrivons l'équation $\Delta\sigma = f(\Delta\varepsilon)$ sous la forme $\Delta\sigma - f(\Delta\varepsilon) = \varphi(\Delta\varepsilon) = \mathbf{0}$, où on travaille en 3D.
- Supposons $\Delta\varepsilon^s$ la valeur de $\Delta\varepsilon$ à l'itération s . La méthode de Newton permet de calculer $\Delta\varepsilon^{s+1}$ avec:

$$\begin{aligned}\Delta\varepsilon^{s+1} &= \Delta\varepsilon^s - \frac{\partial\varphi(\Delta\varepsilon)}{\partial\Delta\varepsilon} \Big|_{\Delta\varepsilon^s} : \varphi(\Delta\varepsilon^s) \\ &= \Delta\varepsilon^s - \frac{\partial\Delta\sigma}{\partial\Delta\varepsilon} \Big|_{\Delta\varepsilon^s} : \varphi(\Delta\varepsilon^s)\end{aligned}\tag{59}$$

- Dans leurs algorithmes internes, les codes de calculs par éléments finis doivent solutionner ce type d'équation à chaque point d'intégration des éléments.
- Alors, une solution par éléments finis en viscoélasticité sera décomposée en plusieurs **incrément**s et chaque incrément nécessitera quelques **itérations**.

Implémentation

Interconversion

Prin. Corres.

Éléments finis

Principe

▷ Imp. générale

Utilisation

→ Les codes de calcul comme ANSYS et ABAQUS permettent de définir des lois de comportement définies par l'utilisateur.

→ Pour ABAQUS, l'utilisateur doit utiliser la sous-routine UMAT et fournir au code de calcul les informations suivantes:

1. $\frac{\partial \Delta \sigma}{\partial \Delta \epsilon}$
2. La valeur de la contrainte à la fin de l'incrément σ^{n+1}
3. Les nouvelles valeurs des variables internes ξ^{n+1} , s'il y a lieu, à la fin de l'incrément

→ Reprenons l'équation (60)

$$\begin{aligned} \sigma^{n+1} &= (\mathbf{L}_1 + \mathbf{L}_2 : \mathbf{W}_2) : \epsilon^{n+1} + \mathbf{L}_2 : \mathbf{W}_1 : \xi^n \\ &= \mathbf{M}_1 : \epsilon^{n+1} + \mathbf{M}_2 : \xi^n \end{aligned} \quad (60)$$

→ En remarquant que $\Delta \sigma = \sigma^{n+1} - \sigma^n$ et que $\Delta \epsilon = \epsilon^{n+1} - \epsilon^n$, on aura que:

$$\Delta \sigma + \sigma^n = \mathbf{M}_1 : (\Delta \epsilon + \epsilon^n) + \mathbf{M}_2 : \xi^n \quad (61)$$

où ici σ^n , ϵ^n et ξ^n sont indépendants de $\Delta \epsilon$ car ils appartiennent au passé.

→ Alors, avec l'équation (61), on aura que:

$$\frac{\partial \Delta \sigma}{\partial \Delta \varepsilon} = \mathbf{M}_1 \quad (62)$$

ce qui illustre que notre technique d'intégration de la loi de comportement est directement compatible avec une formulation par éléments finis.

→ La contrainte à la fin de l'incrément sera donnée par

$$\sigma^{n+1} = \mathbf{M}_1 : \varepsilon^{n+1} + \mathbf{M}_2 : \xi^n$$

→ On peut voir que la contrainte à la fin de l'incrément dépend des variables internes ξ . Ces variables doivent donc être stockées en mémoire et mises à jour à chaque pas de calcul. On avait obtenu précédemment, à l'équation (12) que

$$\xi^{n+1} = \mathbf{W}_1 : \xi^n + \mathbf{W}_2 : \varepsilon^{n+1}. \text{ Alors, à la fin de l'incrément, } \xi^{n+1} \text{ sera calculée avec la valeur de } \varepsilon^{n+1}.$$

→ Nous avons donc là tous les éléments pour implémenter n'importe quelle loi de comportement viscoélastique linéaire, peu importe son degré de symétrie, dans un code de calculs par éléments finis.

Implémentation

Interconversion

Prin. Corres.

Éléments finis

Principe

▷ Imp. générale

Utilisation

Implémentation

Interconversion

Prin. Corres.

Éléments finis

Principe

Imp. générale

▷ Utilisation

- Pour des matériaux isotropes, les codes de calculs ont déjà implémenté des lois de comportement viscoélastiques linéaires. Il est donc possible pour l'utilisateur d'entrer des données isotropes sans avoir lui-même à programmer sa loi de comportement.
- Pour définir un matériau viscoélastique, ANSYS et ABAQUS nécessitent de définir en premier temps un matériau élastique linéaire isotrope. En général, l'utilisateur va entrer un module d'Young E_0 et un coefficient de Poisson ν_0 .
- ANSYS va représenter le module de cisaillement et de compressibilité sous la forme:

$$k(t) = k_0^{\text{ANSYS}} \left[\alpha_{\infty}^{k/\text{ANSYS}} + \sum_{i=1}^N \alpha_i^{k/\text{ANSYS}} \exp \left(-\frac{t}{\tau_i^{k/\text{ANSYS}}} \right) \right]$$

$$\mu(t) = \mu_0^{\text{ANSYS}} \left[\alpha_{\infty}^{\mu/\text{ANSYS}} + \sum_{i=1}^N \alpha_i^{\mu/\text{ANSYS}} \exp \left(-\frac{t}{\tau_i^{\mu/\text{ANSYS}}} \right) \right]$$

k_0 et μ_0 seront déduits de E_0 et ν_0 . L'utilisateur devra fournir à ANSYS les couples (α_i^k, τ_i^k) et $(\alpha_i^{\mu}, \tau_i^{\mu})$.

Implémentation

Interconversion

Prin. Corres.

Éléments finis

Principe

Imp. générale

▷ Utilisation

$$k(t) = k_0^{\text{ANSYS}} \left[\alpha_{\infty}^{k/\text{ANSYS}} + \sum_{i=1}^N \alpha_i^{k/\text{ANSYS}} \exp \left(-\frac{t}{\tau_i^{k/\text{ANSYS}}} \right) \right]$$

$$\mu(t) = \mu_0^{\text{ANSYS}} \left[\alpha_{\infty}^{\mu/\text{ANSYS}} + \sum_{i=1}^N \alpha_i^{\mu/\text{ANSYS}} \exp \left(-\frac{t}{\tau_i^{\mu/\text{ANSYS}}} \right) \right]$$

- Les paramètres α_{∞} sont définis de manière à ce que lorsque $t = \infty$, $k = k_0 \alpha_{\infty}^k$ et $\mu = \mu_0 \alpha_{\infty}^{\mu}$. Les paramètres α_{∞} permettent donc de calculer la diminution de k_0 ou μ_0 lorsque l'on attend très longtemps.
- Les codes calculent $\alpha_{\infty} = 1 - \sum \alpha_i$. Alors, quand $t = 0$, $k = k_0$ et $\mu = \mu_0$. Les modules k_0 et μ_0 représentent donc la réponse instantanée du matériau.
- La définition d'un matériau viscoélastique linéaire dans ABAQUS est très similaire.

→ Dans ce cours, nous avons défini $k(t)$ and $\mu(t)$ comme suit:

$$k(t) = k_0 + \sum_{i=1}^N k_i \exp\left(-\frac{t}{\tau_i^k}\right)$$

$$\mu(t) = \mu_0 + \sum_{i=1}^N \mu_i \exp\left(-\frac{t}{\tau_i^\mu}\right)$$

→ Par correspondance, on a les relations suivantes pour $k(t)$:
(voir démo au tableau)

$$k_0^{\text{ANSYS}} = k_0 + \sum_{i=1}^N k_i \quad , \quad \tau_i^{k/\text{ANSYS}} = \tau_i^k$$

$$\alpha_i^{k/\text{ANSYS}} = \frac{k_i}{k_0^{\text{ANSYS}}} \quad , \quad \alpha_\infty^{k/\text{ANSYS}} = \frac{k_0}{k_0^{\text{ANSYS}}}$$

→ Les mêmes relations s'appliquent pour $\mu(t)$.

Implémentation

Interconversion

Prin. Corres.

Éléments finis

Principe

Imp. générale

▷ Utilisation