

Apport de l'apprentissage par renforcement pour le pilotage d'analyse post-flambage en calcul des structures

E. Marquis^{1,2}, E. Baranger¹, P.-A. Guidault¹, E. Agouzal², T. Douillet-Grellier², M. Courtois²

¹ Université Paris-Saclay, CentraleSupélec, ENS Paris-Saclay, CNRS, LMPS - Laboratoire de mécanique Paris-Saclay, 91190, Gif-sur-Yvette, France. Mail : {elio.marquis,emmanuel.baranger,pierre-alain.guidault}@ens-paris-saclay.fr

² EDF R&D ERMES, Saclay, France. Mail: {eki.agouzal,thomas.douillet-grellier,mathieu.courtois}@edf.fr

Résumé — Les analyses de flambage industrielles sont complexes et requièrent une expertise approfondie. La définition des incréments de longueur d'arc, essentielle à la convergence du solveur implicite et à l'interprétation de la réponse structurelle, est d'abord étudiée à travers une méthode classique utilisée dans `code_aster`. Une approche analytique est ensuite proposée afin d'améliorer la robustesse et la maîtrise de la convergence. Enfin, les limites de ces méthodes sont discutées, conduisant à envisager l'apprentissage par renforcement comme solution innovante pour automatiser la définition des incréments.

Mots clés — analyse post-flambage, sélection de la taille d'incrément, méthode de continuation, apprentissage par renforcement.

1 Introduction

Électricité de France (EDF) est un groupe français chargé de la production d'électricité en France. Une de ses missions est en outre d'y accompagner la transition écologique ainsi que d'assurer la neutralité carbone de la production électrique nationale. En plus de ses missions de transition écologique, EDF est également chargé d'assurer la sûreté, la fiabilité ainsi que l'intégrité des installations de production d'énergie, en particulier au sein des unités de production nucléaires.

La modélisation et la simulation numérique offrent des outils cruciaux pour atteindre ces objectifs. Ils permettent de prendre en compte des effets non linéaires de plus en plus nombreux et variés afin d'affiner les modèles utilisés et de mieux représenter les contraintes que subissent les installations. Dans le cadre de la mécanique des structures, EDF développe le logiciel open source `code_aster` [5]. Ce dernier est un code de calcul éléments finis de qualité industrielle. Avec l'augmentation de la complexité des modèles utilisés, la mise en donnée des simulations devient également de plus en plus ardue pour les ingénieurs d'EDF. Il y a ainsi une forte demande de simplifier la mise en données ainsi que la mise en œuvre des différentes méthodes disponibles.

En particulier, les études de post-flambage posent des problèmes. En effet, ces études nécessitent une première analyse de flambement pour déterminer la charge critique ainsi que les modes de flambement. Une fois ce calcul effectué, il est alors nécessaire de récupérer le degré de liberté d'intérêt, puis une pratique courante consiste à intégrer un défaut géométrique ou matériau afin d'effectuer une étude du comportement post-critique de la structure. Cette étude de post-flambage nécessite également le choix de la méthode de pilotage (méthode de suivi de courbe, aussi appelée méthode de continuation [1]), ainsi qu'une discrétisation temporelle du problème pour assurer la convergence des solveurs non linéaires utilisés tout en décrivant de façon satisfaisante la courbe d'équilibre suivie.

Ces dernières années, l'utilisation de l'intelligence artificielle s'est grandement démocratisée et les nouvelles méthodes associées sont ainsi de plus en plus utilisées dans le domaine scientifique. L'intelligence artificielle permet d'exploiter l'ensemble des données produites par les entreprises au fil des ans et ainsi de faciliter certains processus. EDF cherche donc à intégrer les méthodes d'apprentissage dans ses codes de calcul afin de simplifier leur utilisation ou d'accélérer la résolution de certains problèmes.

Afin d'obtenir ces courbes d'équilibre, dont le treillis présenté Figure 1 est un exemple simplifié, il a fallu choisir la méthode de pilotage (méthode de Crisfield), choisir les degrés de liberté contrôlés, choisir

la liste d'incrémentes permettant d'assurer la convergence. Tout ce processus rend la réalisation d'un calcul de post-flambage complexe, l'utilisateur doit avoir un certain niveau d'expertise pour effectuer tous ces choix. Le but de ce travail est d'essayer de faciliter les choix de l'utilisateur à l'aide de méthodes d'apprentissage. En particulier, on s'intéresse ici à la définition de la liste d'instant de calcul pour en faciliter la réalisation. En effet, la liste d'instant de calcul a une influence sur la qualité des résultats ainsi que sur les capacités de convergence du solveur. Il est donc important, et peu évident, de bien la définir.

Ainsi, dans un premier temps un exemple de pilotage par longueur d'arc d'un calcul de post-flambage est présente. Dans un second temps, la méthode usuelle de définition de la liste d'incrémentes est présentée, puis une méthode basée sur l'analyse est introduite. Les problèmes associés à ces méthodes sont présentés. Enfin, une nouvelle méthode utilisant l'apprentissage par renforcement est introduite afin de déterminer automatiquement la taille des incrémentes de longueur d'arc.

2 Rappels sur les méthodes de continuation – cas du pilotage en longueur d'arc sphérique

On se place ici dans le cadre de la méthode des éléments finis avec une résolution par la méthode de Newton. On s'intéresse à l'équation d'équilibre suivante : on cherche $\mathbf{u} \in \mathbb{R}^{n_{\text{ddl}}}$ et $\lambda \in \mathbb{R}$ tels que

$$\mathbf{F}(\mathbf{u}, \lambda) = \mathbf{f}_{\text{int}}(\mathbf{u}) - \lambda \mathbf{q}_{\text{ext}} = \mathbf{0} \quad (1)$$

où \mathbf{F} est le résidu d'équilibre, \mathbf{u} est le champ de déplacement, n_{ddl} est le nombre de degrés de liberté du système discrétisé, λ est la variable de chargement scalaire, \mathbf{f}_{int} sont les efforts internes et \mathbf{q}_{ext} le vecteur de chargement unitaire. On traite des problèmes quasi-statiques, on dispose donc d'une suite de solutions $(\mathbf{u}^{(k)}, \lambda^{(k)})$ pour k jusqu'à $n-1$ convergés en temps. On cherche donc par un algorithme de Newton le prochain état d'équilibre au n -ième pas de temps, noté $(\mathbf{u}^{(n)}, \lambda^{(n)})$. L'équation (1) devient alors

$$\mathbf{K} \delta \mathbf{u}_i^{(n)} + \mathbf{q}_{\text{ext}} \delta \lambda_i^{(n)} = -\mathbf{F}(\mathbf{u}_{i-1}^{(n)}, \lambda_{i-1}^{(n)})$$

où $\mathbf{K} = \frac{\partial \mathbf{F}}{\partial \mathbf{u}}$ est l'opérateur tangent à l'itéré $i-1$ et sachant que l'itéré $i-1$ n'est, lui, pas à l'équilibre. L'incrément de déplacement à l'itération i s'écrit ainsi [2]

$$\delta \mathbf{u}_i^{(n)} = -\mathbf{K}^{-1} \mathbf{F}(\mathbf{u}_{i-1}^{(n)}, \lambda_{i-1}^{(n)}) + \delta \lambda_i^{(n)} \mathbf{K}^{-1} \mathbf{q}_{\text{ext}}$$

qu'il est possible de réécrire

$$\delta \mathbf{u}_i^{(n)} = \delta \bar{\mathbf{u}}_i^{(n)} + \delta \lambda_i^{(n)} \delta \mathbf{u}_{t,i}^{(n)} \quad (2)$$

avec $\delta \bar{\mathbf{u}}_i^{(n)} = -\mathbf{K}^{-1} \mathbf{F}(\mathbf{u}_{i-1}^{(n)}, \lambda_{i-1}^{(n)})$ et $\delta \mathbf{u}_{t,i}^{(n)} = \mathbf{K}^{-1} \mathbf{q}_{\text{ext}}$. On note de plus $\Delta \mathbf{u}_i^{(n)} = \sum_{k=1}^i \delta \mathbf{u}_k^{(n)}$. C'est cette décomposition qui est utilisée en pratique pour la résolution, elle permet en effet de grandement simplifier les expressions des équations de pilotage.

Un pilotage par longueur d'arc sphérique est utilisé. Dans ce cadre l'équation de pilotage (ou de contrainte) s'écrit [3, 6]

$$\mathcal{C}(\mathbf{u}, \lambda) = \left(\Delta \mathbf{u}_i^{(n)} \right)^{\top} \Delta \mathbf{u}_i^{(n)} - \Delta l^2 = 0.$$

avec Δl^2 l'incrément de longueur d'arc fixé par l'utilisateur. Cette équation représente une hypersphère imposant une longueur de chemin d'équilibre, d'où son nom. Les itérations de la procédure de Newton ont donc lieu sur cette hypersphère. La fonction f dont on cherche une racine par la méthode de Newton correspond au système

$$f(\mathbf{u}, \lambda) = \begin{bmatrix} \mathbf{F}(\mathbf{u}, \lambda) \\ \mathcal{C}(\mathbf{u}, \lambda) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}$$

avec $\mathcal{C}(\mathbf{u}, \lambda)$ l'équation de contrainte, ce qui correspond à $n+1$ équations à $n+1$ inconnues.

Il est possible de remplacer $\Delta \mathbf{u}_i^{(n)}$ par sa décomposition (voir équation (2)) et d'obtenir une équation du

second ordre en $\delta\lambda_i^{(n)}$ ce qui permet de résoudre d'abord C puis F et ainsi de conserver les propriétés de symétrie et de parcimonie de l'opérateur tangent.

$$C(\mathbf{u}, \lambda) = a_1(\delta\lambda_i^{(n)})^2 + a_2\delta\lambda_i^{(n)} + a_3 = 0,$$

avec

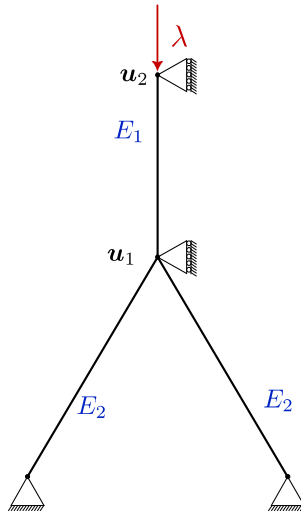
$$a_1 = \left(\delta\mathbf{u}_{i,i}^{(n)}\right)^\top \delta\mathbf{u}_{i,i}^{(n)}, \quad a_2 = 2\delta\mathbf{u}_{i,i}^{(n)}(\Delta\mathbf{u}_i^{(n)} + \delta\bar{\mathbf{u}}_i^{(n)}), \quad a_3 = \left(\Delta\mathbf{u}_{i-1}^{(n)} + \delta\bar{\mathbf{u}}_i^{(n)}\right)^\top (\Delta\mathbf{u}_{i-1}^{(n)} + \delta\bar{\mathbf{u}}_i^{(n)}) - \Delta l^2.$$

Cela permet de déterminer successivement $\delta\lambda_i^{(n)}$ puis $\delta\mathbf{u}_i^{(n)}$. Cependant, comme l'équation est quadratique, deux solutions sont obtenues.

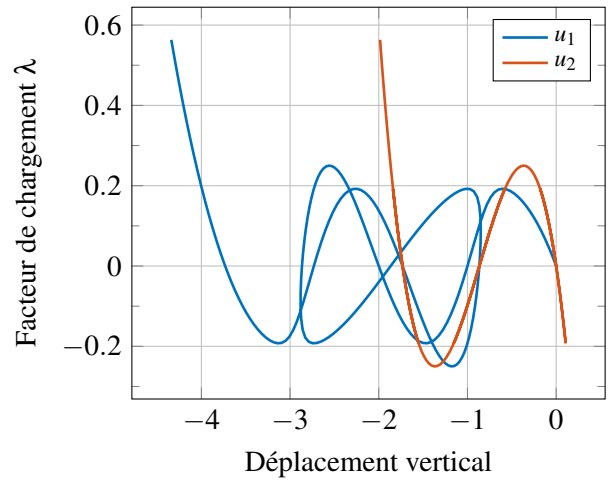
La solution choisie est celle maximisant un cosinus que l'on calcule à partir des deux derniers instants convergés (le tout premier instant est l'état initial ($\mathbf{u}(0) = \mathbf{0}, \lambda(0) = 0$), pour le deuxième instant, la racine positive, i.e. celle augmentant le chargement, est choisie) :

$$\cos\theta = \frac{(\mathbf{u}^{(n)} - \mathbf{u}^{(n-1)})^\top \Delta\mathbf{u}_i^{(n)}}{\Delta l^2} = \frac{(\mathbf{u}^{(n)} - \mathbf{u}^{(n-1)})^\top (\Delta\mathbf{u}_0^{(n)} + \delta\bar{\mathbf{u}}_i^{(n)})}{\Delta l^2} + \delta\lambda_i^{(n)} \frac{(\mathbf{u}^{(n)} - \mathbf{u}^{(n-1)})^\top \delta\mathbf{u}_{i,i}^{(n)}}{\Delta l^2}.$$

Enfin, la méthode de Newton nécessite de faire une prédiction pour la première itération. Classiquement cette prédiction est faite à l'aide de la direction tangente à la courbe. Ici, cette prédiction est faite par extrapolation. Cette méthode consiste à utiliser les deux derniers pas convergés et est présentée dans [7].



(a) Treillis à trois barres



(b) Courbes d'équilibre de la structure

FIGURE 1 – Analyse d'une structure treillis à trois barres en non-linéaire géométrique avec $E_1 = E_2 = 1.0$ et $\Delta l = 0.05$

Ainsi,

$$\begin{pmatrix} \mathbf{u}_1^{(n+1)} \\ \lambda_1^{(n+1)} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_{pred} \\ \lambda_{pred} \end{pmatrix} = \left(1 + \frac{\Delta l^{(n+1)}}{\Delta l^{(n)}}\right) \begin{pmatrix} \mathbf{u}^{(n)} \\ \lambda^{(n)} \end{pmatrix} - \frac{\Delta l^{(n+1)}}{\Delta l^{(n)}} \begin{pmatrix} \mathbf{u}^{(n-1)} \\ \lambda^{(n-1)} \end{pmatrix}$$

Pour le deuxième instant, la prédiction est effectuée de manière tangente.

Ces méthodes peuvent alors être appliquées à l'analyse post-flambage d'un treillis de trois barres (Figure 1) qui illustre la complexité des chemins d'équilibres étudiés.

3 Définition d'une liste d'incréments

Quelques méthodes de définition de la liste d'incréments sont décrites ici. La première méthode développée est celle ayant permis d'obtenir les résultats de la Figure 1b. Une méthode faisant appel à des outils d'analyse est présentée ensuite.

3.1 Méthode usuelle dans le cadre du pilotage par longueur d'arc

L'utilisateur définit une liste d'instants (de longueur d'arc) l_1, l_2, \dots, l_n pour lesquels la solution est calculée. À chaque intervalle est donc associé un incrément de longueur d'arc $\Delta l_1, \Delta l_2, \dots, \Delta l_{n-1}$. Si la procédure de Newton converge rapidement (1 itération), la taille de pas est multipliée par un coefficient β prédéfini et alors

$$\Delta l_{n+1} = \beta \Delta l_n \tag{3}$$

Si la procédure de Newton ne converge pas, la taille de pas est divisée par le coefficient β

$$\Delta l_{n+1} = \frac{\Delta l_n}{\beta} \tag{4}$$

Une fois que le chargement atteint le niveau pour lequel il avait échoué, la taille de pas redevient Δl_n . Cette procédure est illustrée Figure 2 avec $\beta = 2$.

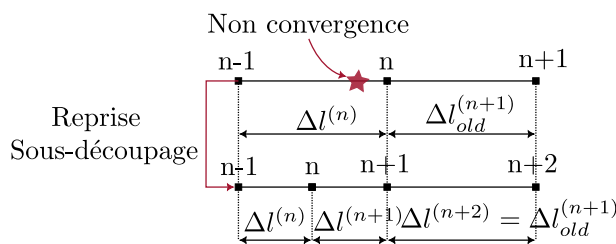


FIGURE 2 – Procédure de sous découpage

3.2 Détermination analytique du rayon de convergence de la méthode de Newton

Une première idée est d'utiliser des outils analytiques afin de voir si garantir le fait que le point de départ soit dans le bassin de convergence de la méthode de Newton permet de garantir la convergence et donc la bonne discrétisation.

Il existe quelques méthodes d'analyse permettant d'estimer le rayon de convergence de la méthode de Newton à partir du point de départ que l'on se donne. On s'intéresse ici à la méthode basée sur le théorème de Kantorovich [8].

Ce théorème permet de calculer un rayon de convergence théorique pour lequel la convergence est assurée. Si la taille de pas est plus grande, la convergence peut avoir lieu, elle n'est juste pas garantie. De plus, cette estimation est effectuée a priori, sans connaître la solution. Ce théorème fournit donc une condition suffisante de convergence locale.

Le théorème de Kantorovich est utilisé afin d'estimer une taille de pas optimale à chaque incrément. À chaque début d'incrément, le solveur fait appel à un estimateur du rayon de Kantorovich afin de choisir la valeur de Δl . Si l'estimateur ne fournit pas de valeur, l'algorithme utilise celle fournie par l'utilisateur (0.07 ici).

Deux solutions sont comparées :

- utilisation d'une taille de pas fixée par l'utilisateur, méthode appelée "Fixée" dans le Tableau 1 ;
- la taille de pas est donnée par l'estimateur, méthode appelée "Kantorovich".

TABLEAU 1 – Comparaison entre les deux méthodes

Méthode	Nombre d'incréments	Nombre d'itérations	Itérations moyennes
Fixée	64	215	3.4
Kantorovich	61	259	4.32

Dans les deux cas, la solution possède le même instant final.

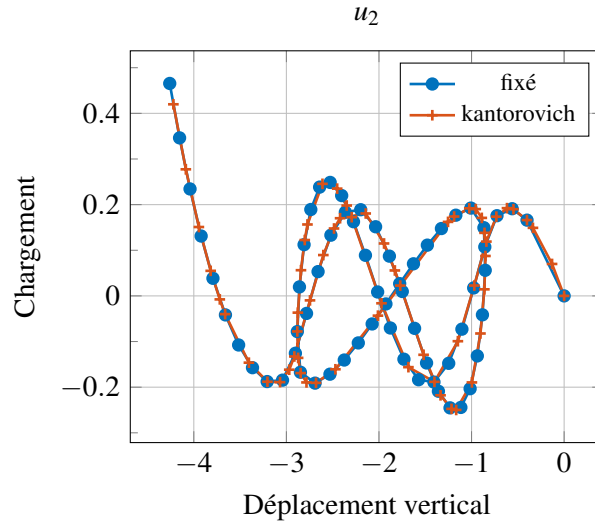


FIGURE 3 – Points convergés pour le cas du treillis de la Figure 1 définis pour une taille de pas fixée à $\Delta l = 0.07$ et une taille de pas déterminé par le critère de Kantorovich

La Figure 3 montre que l'utilisation de la méthode de Kantorovich permet d'augmenter le nombre de points dans les parties non linéaires i.e. à forte courbure. Cependant, le Tableau 1 montre qu'utiliser le rayon donné par la méthode de Kantorovich ne permet pas de diminuer le nombre moyen d'itérations, de plus cela augmente le nombre total d'incrément ainsi que le nombre total d'itérations. Enfin, l'estimation du rayon a un coût de calcul non négligeable (estimation de la constante de Lipschitz du Jacobien, plusieurs estimations du résidu).

Ainsi, bien que le théorème de Kantorovich permette de déterminer une taille de pas garantissant la convergence quadratique, le coût de calcul associé est trop important pour une utilisation sur des problèmes plus complexes.

4 Difficulté de description des courbes d'équilibre à partir de critères de convergence uniquement

Les deux méthodes introduites dans la partie précédente définissent la liste d'incrément permettant de décrire la courbe d'équilibre à partir de critères de convergence de la méthode de Newton. Cependant, il semblerait qu'utiliser uniquement des indicateurs sur la convergence ne soit pas nécessairement une stratégie permettant une description efficace de la courbe.

4.1 Exemple sur une courbe analytique

On considère le cas d'un cercle (courbure constante). L'équation que l'on cherche à résoudre par la méthode de Newton est $f(x) = \sqrt{r^2 - (x-a)^2} - \lambda$. avec $r = 5$, $a = 5$ et $\lambda = 0.9999r$. Le but est de voir s'il existe un sous découpage optimal permettant de minimiser le nombre d'itérations pour aller de $x_0 = 10^{-10}$ jusqu'à la solution $f(x^*) = 0$. Deux essais sont réalisés : un premier cas où aucun découpage n'est effectué, un deuxième cas pour lequel un découpage préalable est effectué.

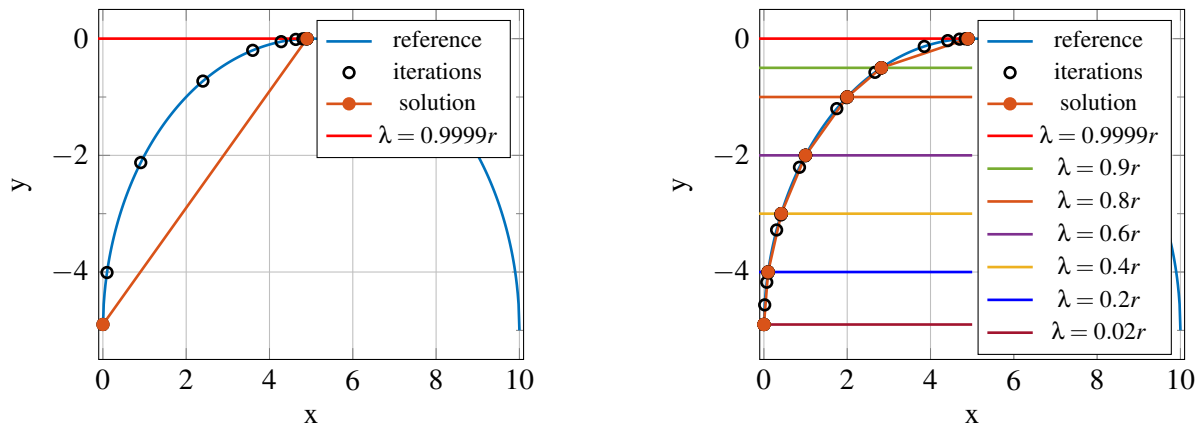


FIGURE 4 – Itérés de la résolution par la méthode de Newton de la fonction $f(x) = 0$ (avec $x_0 = 10^{-10}$) pour une convergence en un pas (gauche) et six pas de calcul. Les lignes horizontales sont les différents λ imposés, les points correspondent aux résultats des sous itérations de Newton.

La Figure 4 montre qu'un découpage optimal pour minimiser le nombre d'itérations semble être de ne pas sous découper et de faire un seul grand pas pour atteindre la solution. Un deuxième découpage optimal serait de découper de manière à ce que chaque itéré calculé précédemment soit solution. Dans un cas la solution obtenue est une droite, dans l'autre on obtient bien un cercle. Le problème de la deuxième solution étant la nécessité de connaître les itérés avant d'effectuer le calcul.

Ainsi, la Figure 4 montre que lorsque la liste d'incrément est définie avec comme seul objectif d'accélérer le temps de calcul en limitant le nombre total d'itérations, il est préférable de faire un unique pas plutôt que de sous-découper. Le problème que cela pose réside dans la représentation de la solution : *toute la non-linéarité du problème étudié a disparu.*

4.2 Exemple sur une courbe d'équilibre mécanique

Désormais, le treillis présenté précédemment (Figure 1) est de nouveau étudié afin de voir l'impact de la taille de pas de longueur d'arc sur la qualité de la solution obtenue. La liste d'incrément est définie par un jeu d'essai erreur afin d'obtenir les plus grands pas possibles permettant la convergence.

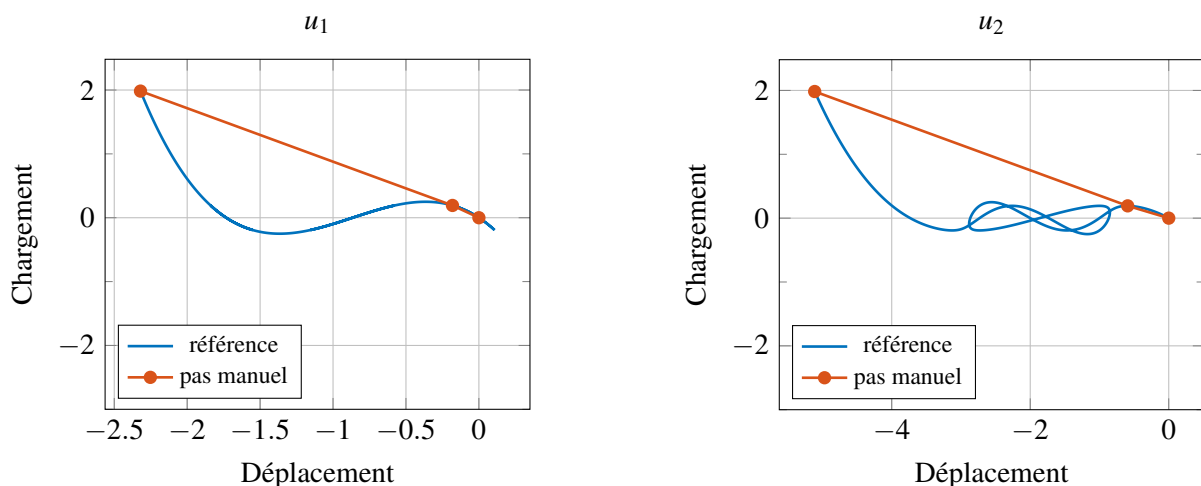


FIGURE 5 – Illustration de la taille de pas maximale pour converger sur la simulation des barres en grandes déformations, 15 itérations maximum, pas de 0.62 puis de 5 pour Crisfield

La Figure 5 montre que la simulation n'a besoin que de deux grands pas pour converger jusqu'au point final. Ce faisant, l'impression donnée est que le comportement de la structure est quasi-linéaire, toutes les non linéarités ne sont pas rendues visibles pour l'utilisateur.

Ainsi, avoir la convergence comme unique critère pour définir la liste d'incrément de longueur d'arc ne permet pas nécessairement d'observer le caractère non linéaire d'une structure, il est nécessaire d'ajouter plus d'informations. Une des questions ainsi soulevée est donc comment bien définir la liste d'instant pour obtenir l'allure satisfaisante d'une courbe d'équilibre.

5 Description d'une courbe d'équilibre via l'apprentissage par renforcement

Plutôt que de laisser l'utilisateur définir la liste, l'idée ici est d'utiliser un agent ayant appris par renforcement (RL) [11] à définir une liste d'incrément permettant de décrire la courbe d'équilibre de manière satisfaisante. La discrétisation de la courbe, et donc la définition de ce qu'est une bonne courbe est laissée à l'agent.

Dans un premier temps, cette méthode est d'abord testée sur des courbes paramétrées. Ainsi, on considère la courbe en deux dimensions définie par

$$\begin{cases} x(t) = a_x t^3 + b_x t^2 + c_x t \\ y(t) = a_y t^3 + b_y t^2 + c_y t \end{cases}, \forall t \in \mathbb{R}^+$$

L'environnement est la courbe paramétrée que l'agent doit parcourir. Le *but* de l'agent est de définir les *bons incréments* Δt afin de placer correctement les *points d'intérêts* : les points pour lesquels la *courbure* est soit *nulle*, soit *infinie*, ou encore les *points de rupture de pente*. Ce sont ces points qui sont *dimensionnants* pour une structure.

Afin de capter ces changements de courbure (notée κ), la fonction $h(t) = -\log_{10}(|\kappa(t)|) + \alpha|\kappa(t)|$ est introduite. L'hyperparamètre α permet d'ajuster l'influence des grandes valeurs de κ . Cette fonction est adaptée des fonctions d'entropie utilisées en théorie de l'information [4].

L'environnement est ainsi la courbe paramétrée que l'agent peut parcourir. Les coefficients a_i, b_i, c_i sont générés aléatoirement. L'état de l'environnement est le triplet $(x(t), y(t), h(t))$. L'espace d'action est un segment $[0, \Delta t_{max}]$ avec Δt_{max} un hyperparamètre.

La récompense est définie comme la somme de trois contributions : une contribution de progression $r_{progress}$, une contribution de pénalité r_{cost} et une contribution de réussite $r_{success}$:

$$r = r_{progress} + r_{cost} + r_{success}$$

La fonction de progression, basée sur le *potential based reward shaping* [9], est définie par

$$r_{progress} = h(t + \Delta t) - \gamma h(t) - \beta |\Delta t|$$

avec γ et β des hyperparamètres. Le premier terme favorise les actions donnant de bons incréments, le deuxième pénalise les actions donnant des incréments trop importants.

La contribution de pénalité est définie par

$$r_{cost} = -\text{cost} \left(1 + \frac{\beta}{\Delta t}\right)$$

dont le but est de pénaliser les épisodes longs, ainsi que les actions donnant de petits Δt .

La contribution de réussite est définie par

$$r_{success} = (\zeta \text{ si } h(t + \Delta t) > h_{min}) + \begin{cases} \text{bonus} * \left(1 - \frac{n_{steps}}{10n_{steps,min}}\right) & \text{si } n_{steps} > n_{steps,min} \text{ et } t_{max} < t \\ -\text{bonus} & \text{sinon si } t_{max} < t \end{cases}$$

dont le premier terme favorise les points d'intérêts (bonus exceptionnel) et le deuxième terme favorise les épisodes terminant après un nombre minimum de pas, tout en pénalisant les épisodes trop rapides.

Un apprentissage est effectué avec $\gamma = 0.99$, $\alpha = 1$, $\beta = 1 - \gamma$, $\text{cost} = 0.01$, $\zeta = 20$, $h_{min} = 9$, $\text{bonus} = 2$, $n_{steps,min} = 5$. L'algorithme A2C de la bibliothèque *Stable-baselines3* est utilisé pour l'apprentissage [10]. A2C est un algorithme de type acteur-critique dont l'utilisation est répandue dans les problèmes de suivi de trajectoire. L'entraînement est effectué durant 2 000 000 d'épisodes. L'agent apprend sur un nombre restreint de courbes (100 différentes) et est testé sur des courbes non vues durant l'entraînement.

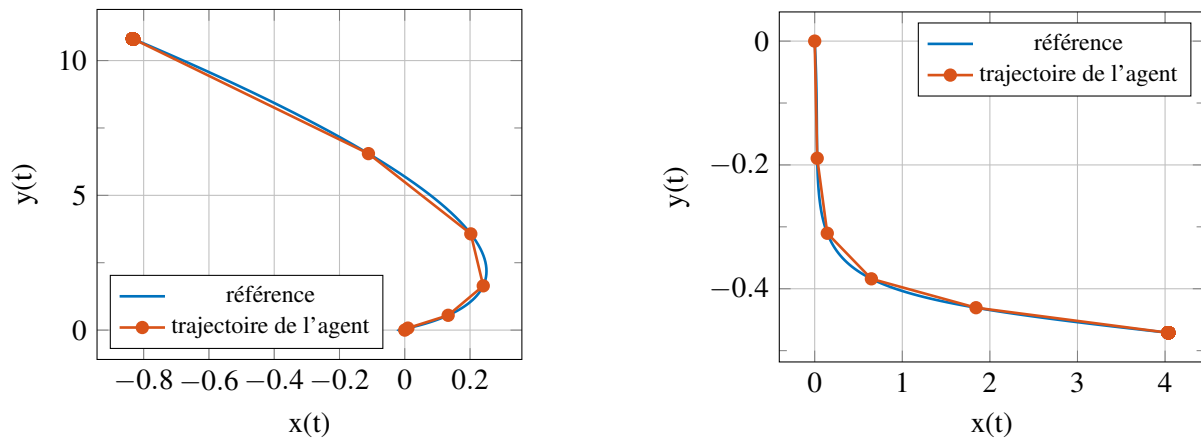


FIGURE 6 – Exemples d'utilisation de l'agent sur des courbes non vues durant l'entraînement

La Figure 6 montre que l'agent arrive bien à placer les points aux points d'intérêt sur des courbes paramétrées. La suite du travail consistera à appliquer ces méthodes à des problèmes de structures en non linéaire géométrique.

6 Conclusion et perspectives

Ce travail a permis de montrer qu'utiliser uniquement des critères de convergence pour définir la liste d'incrément de longueur d'arc lors d'études de post-flambage ne permet pas nécessairement d'obtenir l'allure complète courbes de réponse. L'apprentissage par renforcement semble être une piste prometteuse après des premiers essais sur des courbes paramétrées. Dans la suite, l'approche sera appliquée à des cas de calculs de structures en non-linéaire géométrique. Il conviendra cependant de définir une fonction de récompense adaptée et nécessitant un coût de calcul raisonnable et d'intégrer la nécessité de converger dans le choix de l'incrément de longueur d'arc. De plus, la fonction de récompense, ainsi que l'état de l'environnement, font appel à la courbure au point considéré. Il sera donc nécessaire de disposer de moyens pour estimer cette courbure dans des cas où son calcul n'est plus analytique.

Références

- [1] E. L. Allgower, K. Georg. *Introduction to numerical continuation methods*, SIAM, 2003
- [2] J.-L. Batoz, G. Dhatt. *Incremental displacement algorithms for nonlinear problems*, International Journal for Numerical Methods in Engineering, 1979
- [3] M. A. Crisfield. *A fast incremental/iterative solution procedure that handles snap-through*, Computers & Structures, 1981
- [4] De Boer, P. T., Kroese, D. P., Mannor, S., Rubinstein, R. Y. *A tutorial on the cross-entropy method*, Annals of operations research, 2005
- [5] Électricité de France, *Finite element code_aster, Analysis of Structures and Thermomechanics for Studies and Research*, Open source on www.code-aster.org, 1989-2025
- [6] H.-B ; Hellweg, M.A. Crisfield. *A new arc-length method for handling sharp snap-backs*, Computers & Structures, 1998
- [7] C. Kadapa. *A simple extrapolated predictor for overcoming the starting and tracking issues in the arc-length method for nonlinear structural mechanics*, Engineering Structures, Elsevier, 2021
- [8] L.V. Kantorovich. *On Newton's method for functional equations*, Dokl. Akad. Nauk SSSR, 1948
- [9] A.Y. Ng, D. Harada, S. Russel. *Policy Invariance Under Reward Transformations : Theory and Application to Reward Shaping*, ICML, 1999
- [10] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, N. Dormann. *Stable-baselines3 : Reliable reinforcement learning implementations*, Journal of machine learning research, 2021
- [11] R.S. Sutton, A.G. Barto et al. *Reinforcement learning : an introduction*, MIT press Cambridge, 1998