

# Analyse isogéométrique espace-temps pour la thermique transitoire non linéaire

J. Cornejo Fuentes<sup>1</sup>, G. Sangalli<sup>2</sup>, M. Tani<sup>2</sup>, T. Elguedj<sup>1</sup>, D. Dureisseix<sup>1</sup>, A. Duval<sup>3</sup>

<sup>1</sup> INSA Lyon, CNRS, LaMCoS, UMR5259, Villeurbanne, 69621, France,

(joaquin.cornejo-fuentes, thomas.elguedj, david.dureisseix, arnaud.duvall)@insa-lyon.fr

<sup>2</sup> Università di Pavia, Dipartimento di Matematica, Pavia, 27100, Italy, (giancarlo.sangalli, mattia.tani)@unipv.it

<sup>3</sup> CNRS, INSA Lyon, LaMCoS, UMR5259, Villeurbanne, 69621, France, arnaud.duvall@cnrs.fr

**Résumé** — Cette étude propose une méthode Galerkin espace-temps isogéométrique continue pour résoudre des problèmes non linéaires de transfert thermique, en utilisant des B-splines d'ordre élevé en espace et en temps. Un nouveau préconditionneur, basé sur une diagonalisation rapide et robuste, est introduit, couplé à des solveurs non linéaires (Newton, Picard) pour traiter les matériaux industriels. Les résultats montrent une efficacité supérieure aux méthodes incrémentales classiques, avec des gains de temps de calcul significatifs et une réduction des coûts grâce à des solveurs non linéaires inexacts.

**Mots clés** — Analyse Isogéométrique, espace-temps, thermique transitoire.

## 1 Introduction

Dans ce travail, nous présentons une méthode de Galerkin espace-temps continue en temps à l'aide de l'analyse isogéométrique [14] (IGA) et appliquons celle-ci à la thermique transitoire non linéaire. Ce type d'équations aux dérivées partielles est habituellement résolu par une discrétisation en espace par la méthode des éléments finis ce qui amène à un système d'équations différentielles ordinaire en temps, typiquement résolu à l'aide d'un schéma incrémental en temps. Les méthodes éléments finis espace-temps continues en temps ont été introduites il y a plusieurs décennies [1] en élastodynamique ou conduction thermique [2]. Leur coût numérique prohibitif a cependant limité leur adoption, et les méthodes de Galerkin discontinues en temps leur ont été préférées, celles-ci permettant d'obtenir une résolution incrémentale. Le formalisme unifié espace-temps continu en temps permet cependant d'obtenir une précision en temps et en espace d'ordre élevé, une stabilité inconditionnelle de même que l'absence de restriction sur le pas de temps (cf. [3]). L'approche IGA espace-temps que nous proposons s'appuie sur des travaux récents pour divers types d'équations aux dérivées partielles [4, 5, 6], en prenant en compte des nonlinéarités dans la matrice de comportement. Nous explorons pour cela deux techniques de résolution : la méthode de Newton et la méthode de Picard.

Un des intérêts forts de la méthode proposée est sa capacité à obtenir une solution de très bonne qualité (grâce à des fonctions B-spline d'ordre élevé en espace et en temps) pour un coût numérique extrêmement faible. Cela est rendu possible par l'utilisation de différentes techniques optimisées pour l'usage de fonctions B-spline d'ordre et de continuité élevés. Nous adoptons ainsi les techniques dites à quadrature pondérée WQ [7] qui réduit drastiquement le nombre de points d'intégration, couplées aux approches sans matrice (MF) [8], à l'assemblage par ligne, et de *sum-factorization* [9]. Ceci nous permet de tirer profit de la structure de produit tensoriel et ainsi de développer des algorithmes optimaux. Ceci impose néanmoins d'utiliser des solveurs itératifs, pour lesquels nous avons généralisé le préconditionneur que nous avons développé dans le cas incrémental [10]. Nous étudions également l'influence de la stratégie de résolution itérative entre la boucle interne (solveur linéaire) et la boucle externe (solveur nonlinéaire) pour maîtriser le coût numérique de la méthode, en exploitant les approches dites exactes et inexactes.

## 2 Approche IGA espace-temps de la thermique transitoire nonlinéaire

### 2.1 Forme forte

On considère le domaine espace-temps  $Q = \Omega \times \mathcal{T}$  où  $\Omega \subset \mathbb{R}^d$  représente le domaine spatial en dimension  $d$ , et  $\mathcal{T} = [0, T] \subset \mathbb{R}^+$  l'intervalle de temps. La température  $u$  dépend explicitement de la position  $\mathbf{x} \in \Omega$  et du temps  $t \in \mathcal{T}$ . De plus, on considère des propriétés hétérogènes : la capacité thermique  $\sigma(u, \mathbf{x})$  et la conductivité thermique  $\mathbf{k}(u, \mathbf{x})$  dépendent explicitement de la température  $u$  et de la position  $\mathbf{x}$ . La forme forte du problème de thermique transitoire s'écrit alors :

$$(S) = \begin{cases} \text{Trouver } u : Q \rightarrow \mathbb{R} \text{ tel que :} \\ \sigma \partial_t u - \nabla \cdot (\mathbf{k} \nabla u) = f, & \text{dans } \Omega \times \mathcal{T}; \\ u = 0, & \text{dans } \Gamma \times \mathcal{T}; \\ u = 0, & \text{dans } \Omega \times \{0\}. \end{cases} \quad (1)$$

Nous ferons l'usage des notations suivantes :  $\partial_t$  est la dérivée par rapport au temps  $t$ ,  $\nabla$  le gradient en espace. Par simplicité, nous considérons une source de chaleur  $f \in L^2(\mathcal{T}; H^{-1}(\Omega))$  indépendante de la température  $u$ , des conditions de Dirichlet homogènes et des conditions initiales nulles.

### 2.2 Forme faible

Nous introduisons les espaces de Hilbert

$$\begin{aligned} \mathcal{X} &= \{w \in L^2(\mathcal{T}; H_0^1(\Omega)) \cap H^1(\mathcal{T}; H^{-1}(\Omega)) \mid w(\mathbf{x}, 0) = 0\}; \\ \mathcal{Y} &= \{w \in L^2(\mathcal{T}; H_0^1(\Omega))\}. \end{aligned}$$

La formulation faible est ainsi donnée par

$$(W) = \begin{cases} \text{Trouver } u \in \mathcal{X}, \text{ tel que } \forall w \in \mathcal{Y} : \\ \mathcal{A}(u; u, w) = \mathcal{F}(w); \end{cases} \quad (2)$$

Où les opérateurs  $\mathcal{A}$  and  $\mathcal{F}$  sont donnés par

$$\begin{aligned} \mathcal{A}(s; u, w) &= \int_Q \partial_t u \sigma(s, \mathbf{x}) w dQ + \int_Q \nabla u \cdot [\mathbf{k}(s, \mathbf{x}) \nabla w] dQ; \\ \mathcal{F}(w) &= \int_Q f(\mathbf{x}, t) w dQ. \end{aligned}$$

Après construction des espaces d'approximation  $\mathcal{X}^h \subset \mathcal{X}$  et  $\mathcal{Y}^h \subset \mathcal{Y}$ , avec  $\dim \mathcal{X}_h = \dim \mathcal{Y}_h$ , la formulation de Galerkin est donnée par :

$$(G) = \begin{cases} \text{Trouver } u^h \in \mathcal{X}^h, \text{ tel que } \forall w^h \in \mathcal{Y}^h : \\ \mathcal{A}(u^h; u^h, w^h) = \mathcal{F}(w^h). \end{cases} \quad (3)$$

### 2.3 Forme discrétisée

Dans le cas monodimensionnel, si  $p$  est le degré polynomial et  $n$  le nombre de fonctions B-splines, un *knot-vector* ouvert du domaine  $[0, 1]$  est donné par  $\Xi = \{\xi_1, \dots, \xi_{n+p+1}\}$ , où  $0 = \xi_1 = \dots = \xi_{p+1} < \xi_{p+2} \leq \dots \leq \xi_n < \xi_{n+1} = \dots = \xi_{n+p+1} = 1$ , les *knots* peuvent être répétés  $m$ -fois. Ainsi,  $m$  définit la continuité des B-splines, notée  $r : m \leq p$  and  $r = p - m$ . A partir du *knot-vector*  $\Xi$ ,  $\hat{b}_A$ ,  $A = 1, \dots, n$ , sont les fonctions B-splines monodimensionnelles de degré  $p$  et continuité  $r$  dans l'espace  $\mathbb{S}_r^p = \text{span} \{\hat{b}_{A,p} \mid A = 1, \dots, n\}$  [14]. Dans le cas multidimensionnel, tel qu'en espace-temps, on considère des fonctions qui dépendent de  $d$  variables spatiales et du temps. Ainsi, étant donné  $n_l, p_l$  for  $l = 1, \dots, d$  et  $n_t, p_t$ , on introduit  $d+1$  *knot-vectors* monodimensionnels  $\Xi_l = \{\xi_{l,1}, \dots, \xi_{l,n_l+p_l+1}\}$  pour  $l = 1, \dots, d$  et  $\Xi_t = \{\xi_{t,1}, \dots, \xi_{t,n_t+p_t+1}\}$  et  $\mathbf{p}$  le vecteur contenant les degrés, *i.e.*,  $\mathbf{p} = (p_s, p_t)$ , où  $p_s = (p_1, \dots, p_d)$ . Les B-splines multidimensionnelles sont ainsi définies par produit tensoriel, *i.e.*,

$$\hat{N}_{\mathbf{A}, \mathbf{p}}(\xi_s, \xi_t) = \hat{B}_{\mathbf{A}_s, \mathbf{p}_s}(\xi_s) \hat{b}_{A_t, p_t}(\xi_t), \quad \text{avec} \quad \hat{B}_{\mathbf{A}_s, \mathbf{p}_s}(\xi_s) = \hat{b}_{A_1, p_1}(\xi_1) \dots \hat{b}_{A_d, p_d}(\xi_d), \quad (4)$$

Par simplicité, on considère que  $\Omega$  est décrit par un seul patch à l'aide d'une paramétrisation B-spline  $F_s$ , de même le temps est décrit par une courbe B-spline à l'aide du mapping  $F_t$ , cf. Figure 1.

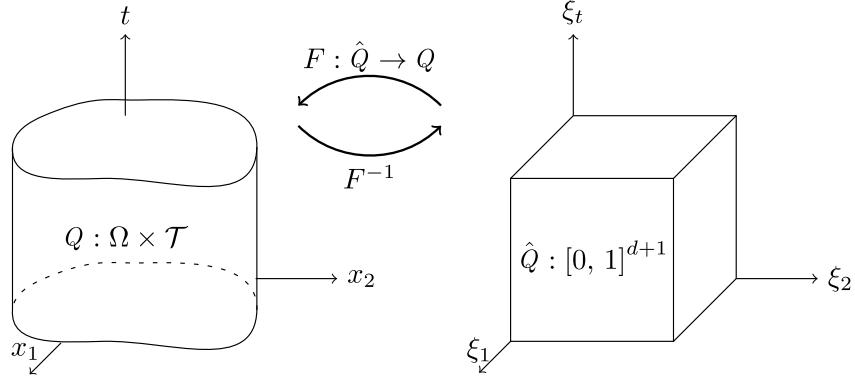


FIGURE 1 – Paramétrisation IGA espace-temps, d'après Langer et al. [4].

Ainsi la forme discrétisée obtenue est donnée par

$$(M) = \begin{cases} \text{Trouver } \mathbf{u} = [u_A]_{A \in \eta} \in \mathbb{R}^{ncp}, \text{ tel que :} \\ \left[ \mathbf{A}(u^h) \right]_{AB} u_B = f_A, \forall A \in \eta; \end{cases} \quad (5)$$

Avec :

$$\begin{aligned} [\mathbf{A}(s)]_{AB} &= \int_Q N_A \boldsymbol{\sigma}(s, \mathbf{x}) \partial_t N_B dQ + \int_Q \nabla N_A \cdot [\mathbf{k}(s, \mathbf{x}) \nabla N_B] dQ; \\ f_A &= \int_Q N_A f dQ. \end{aligned}$$

## 2.4 Résolution nonlinéaire

Dans ce travail, nous utilisons deux approches pour résoudre la nonlinéarité : les méthodes de Newton et Picard. Le résidu nonlinéaire est ainsi donné par

$$\mathbf{r}_A^{\text{NL}}(\mathbf{u}) = f_A - \left[ \mathbf{A}(u^h) \right]_{AB} u_B, \forall A \in \eta, \quad (6)$$

La résolution dans le cas de l'approche de Picard est donnée par

$$\left[ \mathbf{A}(k) \right]_{AB} \left[ u^{(k+1)} \right]_B = f_A, \text{ avec } A \in \eta. \quad (7)$$

Dans le cas de Newton, on obtient

$$\begin{aligned} \left[ u^{(k+1)} \right]_A &= \left[ u^{(k)} \right]_A + [\delta u]_A, \text{ où} \\ [\delta u]_A &= \left[ \mathbf{T}^{(k)} \right]_{AB}^{-1} \left[ \mathbf{r}^{(k)} \right]_B, \forall A \in \eta. \end{aligned} \quad (8)$$

où l'opérateur tangent est donné par

$$\begin{aligned} \left[ \mathbf{T}(u^h) \right]_{AC} &= \left[ \mathbf{A}(u^h) \right]_{AC} + \left[ \frac{\partial \mathbf{A}(u^h)}{\partial u^h} \right]_{AB} u_B = \left[ \mathbf{A}(u^h) \right]_{AC} + \left[ \mathbf{B}(u^h) \right]_{AC} \text{ avec } A, C \in \eta. \\ \left[ \mathbf{B}(u^h) \right]_{AC} &= \int_Q N_A \left( \frac{\partial \boldsymbol{\sigma}}{\partial u^h} \partial_t u^h \right) N_C dQ + \int_Q \nabla N_A \cdot \left( \frac{\partial \mathbf{k}}{\partial u^h} \nabla u^h \right) N_C dQ. \end{aligned} \quad (9)$$

Les deux méthodes de résolution nonlinéaires utilisées ont chacune leurs avantages et inconvénients. Newton permet une convergence quadratique alors que Picard ne converge que linéairement. Cependant Newton nécessite de calculer l'opérateur tangent, et est relativement sensible à la solution initiale et peut donc diverger, ce qui n'est généralement pas le cas de Picard. Étant donné que nous utilisons un solveur

itératif pour résoudre le système linéaire, nous envisageons deux approches possibles. Dans un premier cas, l'approche exacte, consiste à utiliser une tolérance très faible pour le solveur linéaire pour toutes les itérations nonlinéaires. L'approche inexacte propose à contrario de choisir la tolérance du solveur linéaire en fonction de l'état de la résolution nonlinéaire. Lorsque le résidu nonlinéaire est élevé, il n'est pas nécessaire de résoudre avec une très grande précision le système linéaire. Plus on s'approche de la convergence nonlinéaire, plus on va réduire le critère d'arrêt du solveur linéaire de façon à atteindre la convergence asymptotique du solveur nonlinéaire utilisé. Cette adaptation du critère d'arrêt du solveur linéaire est réalisée à l'aide d'un terme de forçage [13].

### 3 Approche sans matrice et à quadrature pondérée - préconditionnement

#### 3.1 Intégration numérique

L'intégration numérique des opérateurs a été traditionnellement traitée en IGA comme en éléments finis, soit à l'aide d'intégration de Gauss sur chaque élément où intervalle entre *knots*. D'autres approches intégrant exactement les opérateurs, mais plus efficaces, ont été proposées ; la quadrature pondérée est celle que nous utilisons ici [7]. Dans le cas monodimensionnel on cherche à calculer l'intégrale

$$\int_0^1 \hat{b}_{A_l}^{(\alpha_l)}(\xi_l) g(\xi_l) d\xi_l,$$

où  $\alpha_l$  est un paramètre qui vaut 1 ou 0 *i.e.*,  $\bullet^{(0)} = \bullet$  et  $\bullet^{(1)} = \partial\bullet/\partial\xi_l$ .  $\hat{b}_{A_l}$  est une fonction B-spline 1D et  $g(\xi_l)$  une fonction qui peut dépendre de la géométrie et des propriétés matériau. Dans l'approche à quadrature pondérée l'intégrale est calculée par :

$$\int_0^1 \hat{b}_{A_l}^{(\alpha_l)}(\xi_l) g(\xi_l) d\xi_l = \sum_{q_l \in Q_l} \omega_{l,A_l,q_l}^{(\alpha_l)} g(\tilde{\xi}_{q_l}). \quad (10)$$

Ainsi les fonctions test  $\hat{b}_{A_l}^{(\alpha_l)}$  font partie du poids d'intégration. Cela signifie que les poids d'intégration dépendent de la fonction B-spline. Néanmoins, cela signifie aussi que la position des points d'intégration n'en dépendent pas. On peut ainsi obtenir des règles d'intégration où le nombre et la position des points d'intégration ne dépendent pas des fonctions B-spline et en particulier de leur degré. Contrairement à l'intégration de Gauss Legendre classique : le nombre et la position des points est donc identique pour des fonctions de bas degré et de haut degré. Le prix à payer est cependant de calculer numériquement les poids d'intégration pour chaque discrétisation.

On remarquera que grâce à la structure de produit tensoriel des B-splines, les règles d'intégration multidimensionnelles s'expriment comme le produit tensoriel des règles d'intégration monodimensionnelles.

#### 3.2 Approche sans matrice

Dans la résolution numérique, nous devons classiquement réaliser des produits matrice-vecteur tels que  $Au$  ou  $T\delta u$ . Dans les deux cas, le calcul de chaque terme et le stockage des matrices n'est pas nécessaire si on dispose d'une routine qui prend en entrée un vecteur  $v$  et retourne le résultat du produit  $Av$  ou  $Tv$ . L'utilisation du produit de Kronecker, et en particulier du produit  $l$ -mode (multiplication d'un tenseur par une matrice) dans un contexte éléments finis ou IGA permet d'exploiter la technique dite de sum-factorization : il n'est pas nécessaire de former explicitement la matrice pour calculer le produit matrice vecteur, cf. [8, 9]. Il est possible d'exploiter de façon particulièrement efficace ces propriétés grâce à la quadrature pondérée puisqu'on dispose alors de règles d'intégration à structure de produit tensoriel, et en réalisant le calcul des termes de la matrice ligne par ligne et non plus élément par élément.

#### 3.3 Préconditionnement

Une des contraintes de l'approche sans matrice est la nécessité d'utiliser un solveur itératif. Il est donc essentiel de construire un préconditionneur efficace tirant parti de la structure de produit tensoriel.

Nous utilisons ici une généralisation du préconditionneur que nous avons proposé dans le cas incrémental [10], celui-ci étant basé sur l'approche dite *Fast-Diagonalization* [11]. Le préconditionneur s'écrit :

$$[P]_{AB} = \int_{\hat{Q}} \hat{N}_A \tilde{c}_\sigma \hat{\partial}_\tau \hat{N}_B d\hat{Q} + \int_{\hat{Q}} \hat{V} \hat{N}_A \tilde{C}_K \hat{V} \hat{N}_B d\hat{Q}, \quad (11)$$

où  $\tilde{C}_K = \text{diag}(\tilde{c}_{K,1}, \dots, \tilde{c}_{K,d})$ , les paramètres scalaires étant calculés par

$$\tilde{c}_\sigma = \int_{\hat{Q}} \hat{\sigma} \det J_{F_s} d\hat{Q} \quad \text{and} \quad \tilde{C}_K = \text{diag} \left( \int_{\hat{Q}} \left( J_F^{-1} \hat{k} J_F^{-\top} \right) \det J_{F_s} \det J_{F_t} d\hat{Q} \right).$$

En exploitant la structure tensorielle des fonctions B-splines multidimensionnelles et les produits de Kronecker on peut obtenir pour un cas 4D espace-temps :

$$P = \tilde{c}_\sigma L_t \otimes M_3 \otimes M_2 \otimes M_1 + M_t \otimes [\tilde{c}_{K,1} M_3 \otimes M_2 \otimes K_1 + \tilde{c}_{K,2} M_3 \otimes K_2 \otimes M_1 + \tilde{c}_{K,3} K_3 \otimes M_2 \otimes M_1]; \quad (12)$$

où  $M_l$  et  $K_l$  sont les matrices de masse et raideur spatiales 1D dans chaque direction,  $M_t$  la matrice de masse 1D temporelle et  $L_t$  une matrice hybride temporelle.

En introduisant la factorisation généralisée des couples  $(K_l, M_l)$  d'une part et la décomposition de Schur généralisée du couple  $(L_t, M_t)$  on peut calculer l'inverse du préconditionneur pour un coût faible et indépendant du degré des B-splines espace-temps.

## 4 Résultats numériques

Tous les résultats présentés ici ont été obtenus à l'aide d'un ordinateur portable équipé de processeurs Intel i7-11850H cadencés à 2,5 GHz et doté de 16 Go de RAM. Les différents tests ont été programmés en Python, version 3.8.10. Cependant, les algorithmes principaux présentés tout au long de l'article, aujourd'hui intégrés au code open-source IGA YETI [12], ont été développés en Fortran90. Ils ont été compilés en bibliothèque statique à l'aide du compilateur GNU Fortran, version 9.4.0, puis appelés en Python via le module *f2py*. Par ailleurs, notre implémentation utilise des fonctions de la bibliothèque LAPACK, version 3.9.0. Dans notre benchmark, la parallélisation n'a pas été prise en compte et tous les calculs ont été restreints à un seul thread de calcul.

On considère le domaine spatial 2D  $\Omega = \{ \mathbf{x} = (x_1, x_2) \mid 0 \leq x_1, 0 \leq x_2, 0.25^2 \leq x_1^2 + x_2^2 \leq 1^2 \}$  et temporel  $\mathcal{T} = [0, 1]$ . Les propriétés matériau sont  $\sigma = 1$  et

$$\mathbf{k}(u) = (3 + 2 \tanh(u/50)) \begin{bmatrix} 1 & 0.5 \\ 0.5 & 2 \end{bmatrix},$$

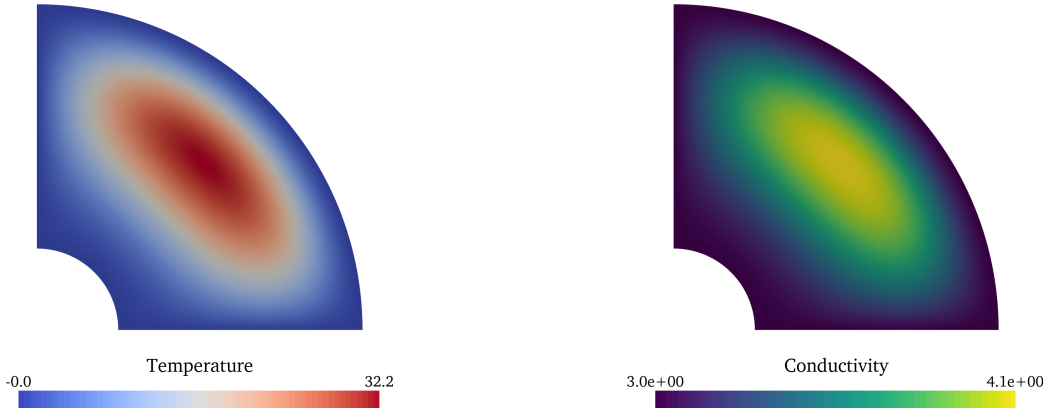
On considère une température initiale nulle et des conditions de Dirichlet homogènes. Une source de chaleur est imposée de façon à obtenir la solution manufacturée ci dessous, cf. Figure 2,

$$u(\mathbf{x}, t) = 50 \tanh(1 - x_1^2 - x_2^2) \sin(\pi(x_1^2 + x_2^2 - 0.25^2)) \sin(\pi x_1 x_2) g(t),$$

avec  $g(t) = \sin(\pi t/2) (1 + 0.75 \cos(3\pi t/2))$ .

Une étude de convergence espace-temps, tant spatiale que temporelle a permis de vérifier l'obtention d'une convergence asymptotique et de comparer celle-ci avec une approche incrémentale classique pour la partie temporelle et notre approche pour la partie spatiale [10].

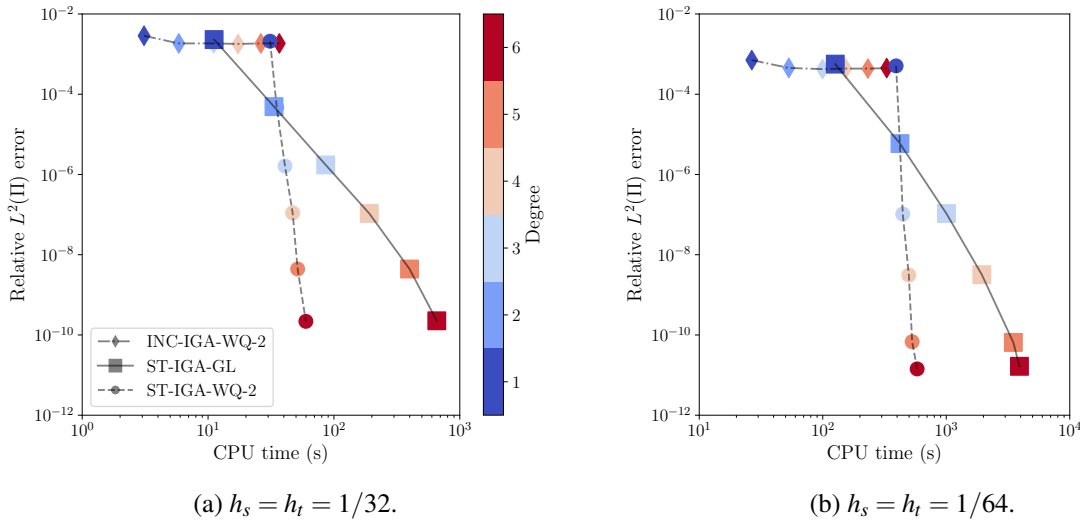
Les Figs. 3a and 3b, comparent les approches incrémentales INC et espace-temps ST soit avec intégration de Gauss-Legendre GL soit la quadrature pondérée WQ. On peut observer que l'approche espace-temps à quadrature pondérée est de loin la plus efficace en terme de ratio temps CPU vs erreur  $L^2$ . On constate également que l'approche espace-temps à quadrature pondérée permet d'obtenir un temps de calcul pratiquement indépendant du degré polynomial. On peut donc obtenir une solution très précise pour un coût numérique extrêmement faible.



(a) Champs de température exacte sur  $\Omega$  à  $T = 1$ .

(b) Conductivité thermique exacte sur  $\Omega$  à  $T = 1$ .

FIGURE 2 – Champs de température et conductivité sur  $\Omega$  à  $T = 1$ .



(a)  $h_s = h_t = 1/32$ .

(b)  $h_s = h_t = 1/64$ .

FIGURE 3 – Précision vs. temps CPU (en secondes) pour les approches incrémentales et espace-temps GL et WQ.

Sur la Fig. 4, l'erreur  $L^2$  espace-temps est tracée en fonction du temps CPU pour différentes discrétisations et degrés. On constate que le coût numérique augmente linéairement avec le nombre d'éléments et est pratiquement indépendant du degré polynomial. Par exemple, si on vise une erreur  $L^2$  de l'ordre de  $10^{-3}$ , on peut choisir un degré  $p_s = p_t = 6$  avec un maillage de  $4^3$  éléments ou un degré  $p_s = p_t = 1$  et  $64^3$  éléments. La différence de temps CPU entre les deux calculs est de l'ordre de 500. Si la précision recherchée est de  $10^{-9}$  l'écart en temps CPU entre  $p_s = p_t = 5$  et  $32^3$  éléments d'une part et l'approche incrémentale (soit  $p_t = 1$ ) et des éléments finis de degré 1 est de l'ordre de  $10^8$  ! Ceci démontre que l'approche ST-IGA-MF-WQ est très largement supérieure, tant en terme de précision que de temps CPU, à une approche incrémentale éléments finis de bas degré.

La Fig. 5a illustre l'évolution de la norme relative du résidu non linéaire au fil des itérations. Comme prévu, les méthodes « exactes » conservent un taux de convergence optimal, contrairement aux méthodes « inexactes ». La méthode de Picard inexacte retrouve une convergence linéaire optimale après 3 itérations, tandis que Newton inexact passe d'une convergence linéaire à superlinéaire, sans atteindre le quadratique. Pour ce problème, le choix des méthodes inexactes a un impact modéré sur le nombre d'itérations non linéaires pour un résidu similaire. En pratique, avec un solveur direct, les méthodes inexactes augmenteraient le coût numérique. Mais avec des solveurs itératifs linéaires, c'est le nombre total d'itérations qui compte. La Fig. 5b montre que Picard et Newton inexacts convergent en environ 20 itérations linéaires, soit 20% à 50% de moins que leurs versions exactes. Cela confirme qu'il est inutile de résoudre précisément le problème linéarisé dès les premières itérations. Les Fig. 5c et Fig. 5d analysent l'erreur en norme  $L^2$  espace-temps. Après quelques itérations, l'erreur ne diminue plus, malgré la réduction du

résidu. Cela s'explique par la négligeabilité de l'incrément face à l'erreur de discrétisation. Conclusion : le critère d'arrêt basé sur la norme euclidienne du résidu n'est pas adapté aux systèmes non linéaires issus de Galerkin. D'autres normes seraient plus pertinentes, mais cela dépasse le cadre de ce travail.

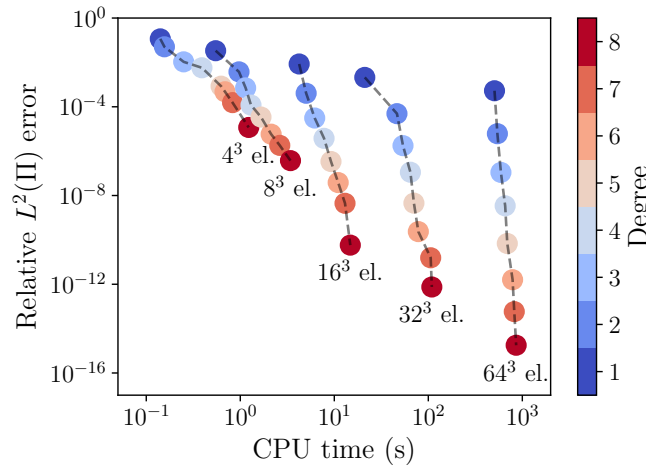
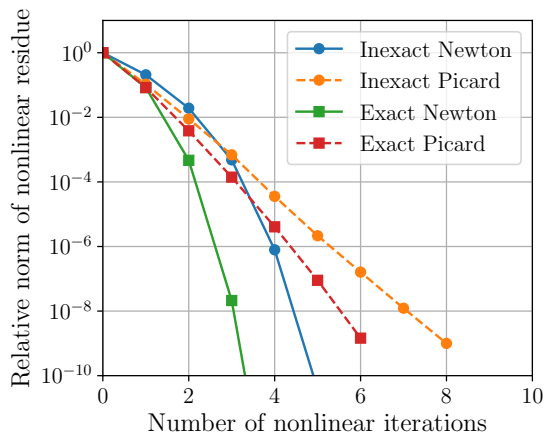
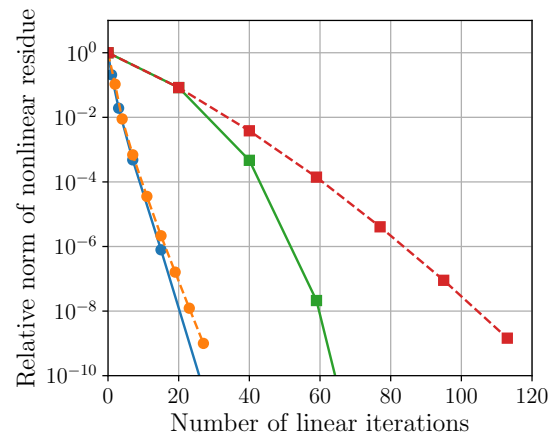


FIGURE 4 – Précision vs. temps CPU (en secondes) pour ST-IGA-MF-WQ pour différents maillages.

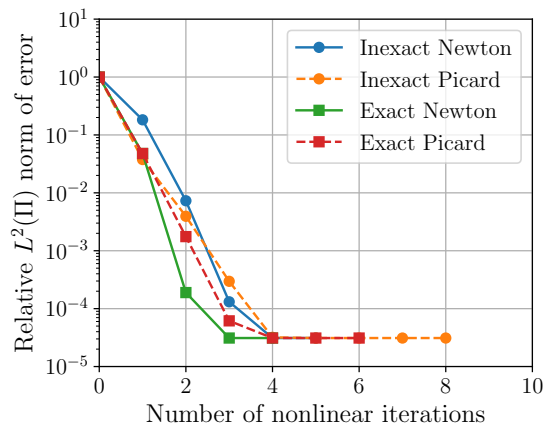
On compare ensuite la performance de solveurs nonlinéaires Newton et Picard avec les approches exactes et inexactes.



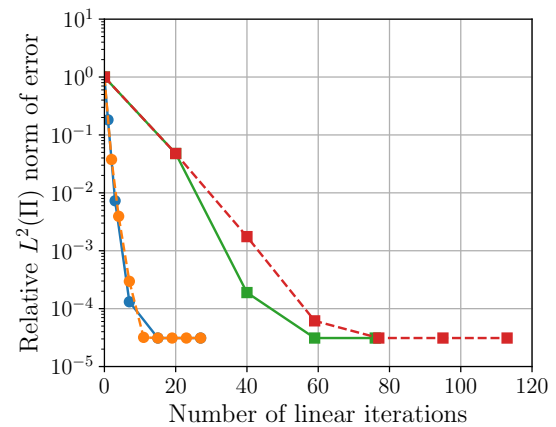
(a) Norme du résidu NL vs nb d'itérations NL.



(b) Norme du résidu NL vs nb d'itérations linéaires.



(c) Norme du résidu NL vs nb d'itérations NL.



(d) Norme du résidu NL vs nb d'itérations linéaires.

FIGURE 5 – Comparaison des solveurs nonlinéaires pour  $p = 3$  et  $h = 1/16$ .

## 5 Conclusions

Dans ce travail, nous avons développé une méthode espace-temps isogéométrique de Galerkin continue en temps pour résoudre des problèmes non linéaires de transfert thermique transitoire, en ciblant la réduction des coûts liés à l'augmentation de la dimensionnalité. Pour optimiser le temps CPU et la mémoire, nous avons utilisé les approches sans matrice à quadrature pondérée pour des opérations matrice-vecteur efficaces ainsi qu'un préconditionneur de type *Fast-Diagonalization*, robuste et facile à implémenter dans un code IGA.

Nos tests numériques montrent que l'approche espace-temps utilisant des fonctions B-spline de degré et continuité élevées, combiné à MF-WQ surpasse les méthodes traditionnelles, malgré une dimension accrue. Les méthodes ST optimisées sont plus performantes que les approches incrémentales et leur temps CPU est indépendant du degré polynomial. Enfin, l'approche « inexacte » réduit jusqu'à 50% le coût numérique, malgré une convergence sous-optimale. Ces techniques offrent un cadre robuste et efficace pour résoudre des problèmes complexes de transfert thermique, atteignant l'objectif principal de l'étude.

## Références

- [1] J. T. Oden, *A general theory of finite elements. I. Topological considerations*, International Journal for Numerical Methods in Engineering, vol. 1, no. 2, pp. 205–221, 1969
- [2] J. C. Bruch Jr and G. Zvoloski, *Transient two-dimensional heat conduction problems solved by the finite element method*, International Journal for Numerical Methods in Engineering, vol. 8, no. 3, pp. 481–494, 1974
- [3] U. Langer and O. Steinbach, *Space-time methods : Applications to partial differential equations*, De Gruyter, Berlin, Boston, 2019.
- [4] U. Langer, S. E. Moore, and M. Neumuller, *Space-time isogeometric analysis of parabolic evolution problems*, Computer Methods in Applied Mechanics and Engineering vol. 306, pp. 342–363, 2016.
- [5] M. Montardini, M. Negri, G. Sangalli, and M. Tani, *Space-time least-squares isogeometric method and efficient solver for parabolic problems*, Mathematics of Computation, vol. 89, no. 323, pp. 1193–1227 2020.
- [6] G. Loli, M. Montardini, G. Sangalli, and M. Tani, *An efficient solver for space-time isogeometric Galerkin methods for parabolic problems*, Computers and Mathematics with Applications, vol. 80, no. 11, pp. 2586–2603, 2020.
- [7] F. Calabro, G. Sangalli, and M. Tani, *Fast formation of isogeometric Galerkin matrices by weighted quadrature*, Computer Methods in Applied Mechanics and Engineering, vol. 316, pp. 606–622, 2017.
- [8] G. Sangalli and M. Tani, *Matrix-free weighted quadrature for a computationally efficient isogeometric k-method*, Computer Methods in Applied Mechanics and Engineering, vol. 338, pp. 117–133, 2018.
- [9] P. Antolin, A. Buffa, F. Calabro, M. Martinelli, and G. Sangalli, *Efficient matrix computation for tensor-product isogeometric analysis : The use of sum factorization*, Computer Methods in Applied Mechanics and Engineering, vol. 285, pp. 817–828, 2015
- [10] J. Cornejo Fuentes, T. Elguedj, D. Dureisseix, and A. Duval, *A cheap preconditioner based on fast diagonalization method for matrix-free weighted-quadrature isogeometric analysis applied to nonlinear transient heat transfer problems*, Computer Methods in Applied Mechanics and Engineering, vol. 414, p. 116 157, 2023.
- [11] G. Sangalli and M. Tani, *Isogeometric preconditioners based on fast solvers for the Sylvester equation*, SIAM Journal on Scientific Computing, vol. 38, no. 6, A3644–A3671, 2016
- [12] A. Duval, T. Hirschler, J. Cornejo Fuentes, M. Guerder, and T. Elguedj, *YETI : Yet another iga code*, <https://lamcosplm.insa-lyon.fr/projects/yeti>, 2023.
- [13] G. Sangalli and M. Tani, *Isogeometric preconditioners based on fast solvers for the Sylvester equation*, SIAM Journal on Scientific Computing, vol. 38, no. 6, A3644–A3671, 2016
- [14] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs, *Isogeometric analysis : CAD, finite elements, NURBS, exact geometry and mesh refinement*, Computer Methods in Applied Mechanics and Engineering, vol. 194, no. 39, pp. 4135–4195, 2005.