

Etude d'une méthode de compression pour accélérer la résolution des équations de Maxwell en 2D par la méthode FDTD

T. Volpert¹, O. Kaya², M. Khouyaali², J.-R. Poirier³, A. Torri², A. Tur⁴

¹ONERA/DEMR Université de Toulouse F-31055, Toulouse - France

²LISN, Université Paris-Saclay, CNRS, Gif-sur-Yvette, France,

³LAPLACE, Université de Toulouse, CNRS, INPT, UPS, Toulouse, France,

⁴IRT Saint Exupéry, Toulouse, France,

Email : thibault.volpert@onera.fr

Résumé—Dans ce travail préliminaire, nous nous proposons d'étudier une méthode de compression tensorielle afin d'accélérer les temps de calculs lors d'une simulation FDTD 2D en électromagnétisme.

I. INTRODUCTION

La simulation de problèmes électromagnétiques peut se faire par de nombreuses méthodes numériques (différences finies, volumes finis, équations intégrales, éléments finis) qui présentent chacune leurs avantages et leurs inconvénients selon le type de problème abordé. Une des méthodes les plus utilisées est la méthode FDTD (Finite Différence Time Domaine) notamment grâce à sa simplicité de mise en oeuvre et son efficacité. Lorsque la taille du problème (au sens électromagnétique, i.e., lorsque la fréquence et le volume de calcul) augmente, les temps de simulation peuvent devenir prohibitifs. Dans ce travail préliminaire, nous nous proposons d'étudier une méthode de compression basée sur une représentation tensorielle des champs électriques et magnétiques ainsi que des opérateurs qui les lient entre eux.

II. EQUATIONS DE MAXWELL ET FDTD

Considérons les équations de Maxwell 2D en mode TE dans un repère orthonormé cartésien :

$$\begin{cases} \frac{\partial H_z}{\partial t} = \frac{1}{\mu} \left[\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \right] \\ \frac{\partial E_x}{\partial t} = \frac{1}{\varepsilon} \left[\frac{\partial H_z}{\partial y} - \sigma E_x \right] \\ \frac{\partial E_y}{\partial t} = \frac{1}{\varepsilon} \left[-\frac{\partial H_z}{\partial x} - \sigma E_y \right] \end{cases} \quad (1)$$

Une approximation différences finies classique [6] sur une grille à pas constant, peut s'écrire ainsi sous forme matricielle :

$$\begin{cases} H_z^{n+1/2} = H_z^{n-1/2} + A(E_x^n + E_y^n) + V_x \\ E_x^{n+1} = B E_x^n + C H_z^{n+1/2} \\ E_y^{n+1} = D E_y^n + G H_z^{n+1/2} \end{cases} \quad (2)$$

Les valeurs des champs tangentiels aux bords du domaine sont nulles. Le terme V_x représente une source ponctuelle.

Les champs E et H étant représentés par des vecteurs, nous voyons apparaître pour chaque itération temporelle

des produits matrice-vecteur faisant intervenir les vecteurs H_z, E_x, E_y et les matrices A, B, C, D, G .

L'idée de base ici est de compresser toutes ces données d'algèbres afin de réduire le volume de calcul.

III. FORMAT TENSOR-TRAIN

Nous présentons dans cette section le format tensoriel de rang faible que nous avons utilisé pour réduire la taille des différents vecteurs et matrices du problème.

A. Généralités

Le format Tensor-Train (TT) [1] consiste à écrire un tenseur $\mathcal{T} \in \mathbb{K}^{n_1 \times \dots \times n_d}$ ($\mathbb{K} = \mathbb{R}$ ou \mathbb{C}) de dimension d , comme une chaîne de petits tenseurs de dimension trois selon la formule écrite sous forme matricielle :

$$\mathcal{T}(i_1, \dots, i_d) = \underbrace{G_1[i_1]}_{r_0 \times r_1} \underbrace{G_2[i_2]}_{r_1 \times r_2} \dots \underbrace{G_d[i_d]}_{r_{d-1} \times r_d}$$

avec

- G_i : les TT-cores (matrices $r_{k-1} \times r_k$ avec $r_0 = r_d = 1$)
- r_i : les rangs TT de chaque core
- $r = \max(r_i)$: rang maximal des cores

La complexité du stockage est $\mathcal{O}(dnr^2)$ pour un tenseur avec $\mathcal{O}(n^d)$ éléments. Si r est petit, le tenseur est alors dit de bas rang et sa représentation TT ne demande pas beaucoup d'éléments et les opérations algébriques seront accélérées.

Le même format est donné pour représenter un opérateur matriciel dans $\mathbb{R}^{n^d \times n^d}$ qui est représenté par un tenseur \mathcal{A} de dimension $2d$ qui couple les éléments (i_n, j_n) pour $n = 1, \dots, d$ à partir du couple (i, j) .

La représentation TT de ce tenseur a la forme

$$\mathcal{A}(i_1, \dots, i_d; j_1, \dots, j_d) = \underbrace{M_1[i_1, j_1]}_{r_0 \times r_1} \underbrace{M_2[i_2, j_2]}_{r_1 \times r_2} \dots \underbrace{M_d[i_d, j_d]}_{r_{d-1} \times r_d}$$

ou $M_k[i_k, j_k]$ est une matrice de taille $r_{k-1} \times r_k$.

Le format TT permet de réaliser les opérations algébriques (addition, produit matrice-vecteur, produit scalaire, etc.) de façon efficace grâce à la réduction du stockage.

B. Quantization et format QTT

Si certains problèmes physiques peuvent se formuler directement en termes de tenseurs, ce n'est pas le cas de celui que nous considérons ici. Afin de "tensoriser" le problème, nous considérons une technique appelée quantization [2],

qui permet de convertir un vecteur ou une matrice dans une représentation tensorielle.

Pour un vecteur $x \in \mathbb{R}^I$ et $I, d \in \mathbb{N}$ tels que

$$I = I_1 I_2 \dots I_d \text{ avec } I_k \in \mathbb{N} \text{ et } k = 1, \dots, d$$

la transformation $i = i_1 + i_2 I_1 + \dots + i_d I_1 I_2 \dots I_{d-1}$, permet de transformer x en un tenseur \mathcal{X} de dimension d tel que

$$x_i = \mathcal{X}(i_1, \dots, i_d) \text{ avec } i_k = 0, 1, \dots, I_k - 1$$

Cette transformation permet également de définir une TT-matrice de $A \in \mathbb{R}^{I \times I}$. Cette représentation est appelée Quantized Tensor-Train (QTT).

Dans notre approche, les matrices A, B, C, D, G , constantes, seront écrites sous forme QTT une fois pour toute en début de programme. Les vecteurs H_z, E_x, E_y étant variables dans le temps, ils sont recomprimés à chaque itération. L'efficacité de la méthode proposée dépend du taux de compression de ces memes vecteurs qui conditionnent la vitesse des produits matrice-vecteur.

C. Construction de la matrice en format TT

1) *TT-cross*: L'algorithme TT-cross est une procedure itérative qui partant d'un tenseur initial donné \mathcal{X} , de rang fixé, construit une approximation $\tilde{\mathcal{X}}$ obtenu par iterations successives. C'est une technique à priori qui nécessite seulement une fonction f qui calcule quelques éléments de \mathcal{X} à la volée, i.e.,

$$\mathcal{X} = f(i_1, \dots, i_d)$$

pour construire le tenseur approché $\tilde{\mathcal{X}}$ directement sous la forme TT de bas rang [4] avec une précision $\epsilon > 0$ soit $\|\tilde{\mathcal{X}} - \mathcal{X}\|_F \leq \epsilon \|\mathcal{X}\|_F$ ou $\|\cdot\|_F$ est la norme de Frobenius et $\epsilon > 0$ la précision désirée. La procédure nécessite $O(Ndr^2)$ évaluations de f par itération avec r est le rang TT.

Moins précis que l'algorithme TT-SVD [3] (non utilisable ici en pratique), TT-cross surestime généralement le rang du tenseur et il est ensuite nécessaire d'utiliser une opération de recompression à posteriori appelée TT-rounding [4].

Ainsi un vecteur ou une matrice peut être construite directement en format QTT sans construction de la matrice pleine. Pour ce travail nous avons utilisé la librairie ttpy, la version Python de la TT-Toolbox [5]. Bien que TT-cross puisse être utilisé, il est cependant préférable ici d'en effectuer une construction directe pour les matrices .

2) *Construction "directe"*: Chaque matrice (A, B, C, D, G) consiste en plusieurs diagonales constantes (sauf pour les colonnes correspondants aux bords du domaine). Cela permet de les former directement car une matrice avec une seule diagonale constante admet une construction explicite de rang 2 en format TT [2]. Nous exploitons cette brique pour former chaque diagonale et utiliser ensuite l'arithmétique en format TT (addition suivie par rounding) [1] pour former l'opérateur final. Afin de mettre à 0 les bords du domaine, nous construisons un vecteur en format TT de rang 2 qui vaut zéro aux bords du domaine et un partout ailleurs et à chaque application de l'opérateur, nous effectuons un produit Hadamard en format TT [1] suivi par TT-rounding. Cela permet de garder l'opérateur simple et de bas rang.

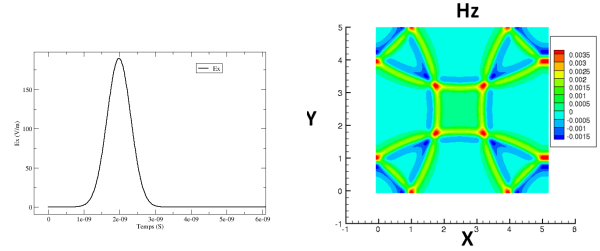


FIGURE 1. Champ électrique temporel et champ magnétique 2D à it=1000

IV. RÉSULTATS NUMERIQUES

Les tests de simulations ont été effectués à l'aide de la librairie ttpy, la version Python de la TT-Toolbox [5]. Conciérons un domaine de calcul de dimension $N_{i+1} * N_{i+1} = 2^d$. Nous placons la source ponctuelle V_x , de type Gaussienne au centre du domaine de calcul. Observons alors l'évolution des temps de calculs en fonction de d et du nombre d'itérations temporelles pour la méthode TT et pour une implémentation classique du schémas de Yee [6]. L'implémentation classique se comporte comme attendu, le temps de calcul dépend linéairement du nombre d'inconnues et du nombre d'itérations. Le temps de calcul pour l'approche TT dépend du contenu des champs électriques et magnétiques et donne de très grandes accélérations lorsque l'on agrandit le domaine pour un nombre d'itérations fixé.

| d | 8 | 9 | 10 | 11 | 12 | 13 |
|--------------|----|----|----|-----|-----|------|
| Exacte (sec) | 2 | 7 | 31 | 117 | 449 | 1809 |
| TT (sec) | 22 | 44 | 69 | 85 | 93 | 102 |

TABLE I
TEMPS DE CALCULS NSTEP=1000.

Les résultats obtenus montrent une bonne réduction du stockage mémoire avec cependant une augmentation du rang et remplissage lorsque le signal se complexifie. La performance en terme de temps de calculs est donc dépendante des paramètres du problème considéré. Ces premiers résultats nous permettent d'espérer un gain significatif en temps calcul pour un certain type de problème plutôt basse fréquence qui par nature sont sur maillés pour tenir compte des géométries impliquées.

RÉFÉRENCES

- [1] I. V. Oseledets. 2011. "Tensor-Train Decomposition" *SIAM J. Sci. Comput.*, vol 33, no 5 September 2011, pp 2295-2317.
- [2] Khoromskij, B. "O (d log N)-Quantics Approximation of N - d Tensors in High-Dimensional Numerical Modeling" *Constructive Approximation*, vol 34, 2009.
- [3] Ivan Oseledets, Eugene Tyrtyshnikov, "TT-cross approximation for multidimensional arrays" *Linear Algebra and its Applications*, Volume 432, Issue 1, 2010, Pages 70-88.
- [4] Sergey V. Dolgov and Dmitry V. Savostyanov. "Alternating Minimal Energy Methods for Linear Systems in Higher Dimensions". *SIAM Journal on Scientific Computing* vol 36, no 5, pp. A2248-A2271, 2014.
- [5] I. V. Oseledets, S. Dolgov, V. Kazeev, D. Savostyanov, O. Lebedeva, P. Zhlobich, T. Mach, and L. Song. *TTToolbox*. <https://github.com/oseledets/TT-Toolbox>.
- [6] K.S. Yee *Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, IEEE trans. Antennas Propag., AP-14 : 302-307, 1966.*