Toolbox harmonique : V0.1

May 9, 2022

I. Notations et convention

A. *Représentation harmonique, phaseurs et évaluation temporelle d'une matrice temps périodique*

Soit $T \in \mathbb{R}^{*+}$, $M(t) \in \mathbb{C}^{n_1 \times n_2}$ T-périodique

On définit la *short time fourier transform* de $M(\cdot)$ par la séquence :

$$\mathcal{F}_T\left(M\right) = \left(\mathcal{M}_k\right)_{k \in \mathbb{Z}}\,;\, \mathcal{M}_k \in \mathbb{C}^{m,n} \tag{1}$$

où $\mathcal{M}_k$ est donné par la formule :

$$\mathcal{M}_k = \frac{1}{T}\int_0^T M(\tau) e^{-jk\frac{2\pi}{T}\tau} \mathrm{d}\tau \tag{2}$$

On note $((\mathcal{M}|_m)_k)_{k \in [-m,m]}$ la troncature à l'ordre m de la séquence $(\mathcal{M}_k)$.

On représente alors par un tableau à trois entrée (formellement un tenseur) $\mathcal{M}|_m : \mathfrak{M}$, dont la troisieme dimension stocke les phaseurs de $M|m$.

$$\begin{cases} \mathfrak{M}(i,j,m+1+k) = \frac{1}{T}\int_0^T M_{i,j}(\tau) e^{-jk\frac{2\pi}{T}\tau} \mathrm{d}\tau \\ (i,j) \in [1:n_1] \times [1:n_2], k \in [-m:m] \end{cases} \tag{3}$$

Ainsi sous MATLAB, on utilisera les 3D-array pour le stockage des phaseurs d'une matrice périodique.

L'évaluation temporelle de la matrice $M(t)$ se calcule alors grace au produit contracté mode-3

$$\begin{cases} (e_i(t))_{i \in [1:2m+1]} \in \mathbb{C}^{2m+1} \Leftrightarrow e_{m+1+k}(t) = e^{jk\frac{2\pi}{T}}t \\ M(t) = \mathfrak{M}\,\overline{\times_3}\, e = \sum_{l=1}^{2m+1} \mathfrak{M}(:,:,l) e_l(t) \end{cases} \tag{4}$$

sous MATLAB la fonction `tensorprod`, disponible depuis la 2022ra permet d'exprimer le produit de 2 tenseur et à fortiori le produit contracté en précisant les dimensions à faire coincider.

```
T=1 %sec

Mph=zeros(2,2,3) %M tronqué à l'ordre 1 (3-1)/2=1
Mph(:,:,1) =[1/2 1/2*1i; -1/2+1i 0]; %phaseur d'ordre -1
Mph(:,:,2) =[0 2 ; -1 -3]; %phaseur d'ordre 0
Mph(:,:,3) =conj(Mph(:,:,1)); %phaseur d'ordre 1


%M(t) à un instant donné
t=0.1; %sec
ei = exp(1i*2*pi/T*(-1:1)'*t); %vecteur e à t fixé, dim1 <=> les ejkw
tensorprod(Mph,ei,3,1)

%fonction anonyme d'évaluation
M =@(t,T) tensorprod(Mph,exp(1i*2*pi/T*(-1:1)'*t),3,1);
M(t,T)

%M(t) evalué sur un un array de temps :
t=0:0.01:1;
eit=exp(1i*2*pi/T*(-1:1)'*t); %array e-temps <=> dim1 les ejkw, temps en dim 2
Mt=tensorprod(Mph,eit,3,1) %est un 3D array dont Mt(:,:,k) est M(t(k))
M(t,T)

figure(1)
plot(t,reshape(Mt,4,[],1)); %on applatit la matrice pour observer les phaseurs
```

A noter la convention suivantes : quelque soit la dimention qui stocke les phaseurs d'un signal (dim 1 pour la fourier d'un scalaire, ou la base temporelle $e(t)$, dim3 pour les matrices...), les m premiers coeeficients stockes les phaseurs négatifs.

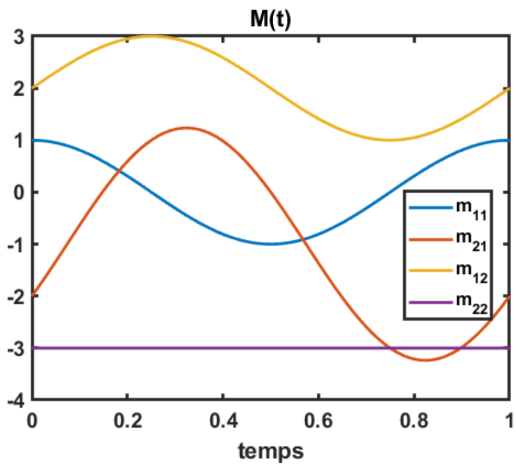

Fig. 1: Coeeficients de M évalués grace à matlab et au calcul tensoriel contracté

B. *Produit de deux matrices*

Si A, B sont deux matrices T-périodiques, AB est T périodique. Alors le phaseurs de AB s'exprime par la convolution matricielle :

$$AB_k = \sum_{l \in Z} A_l B_{k-l} \tag{5}$$

On suppose désormais que l'on travaille avec des matrices tronquées à l'ordre m, et leur représentation 3D $\mathfrak{A}$ et $\mathfrak{B}$:

$$\begin{aligned}
\forall k \in [-m:m] \quad AB_k &= \sum_{l \in [-m:m]} A_l B_{k-l} && (6)\\
&= \sum_{\begin{cases} -m \le l \le m \\ -m \le k-l \le m \end{cases}} \mathfrak{A}(:,:,m+1+l)\mathfrak{B}(:,:,m+1+k-l) && (7)\\
&= \sum_{\begin{cases} -m \le l \le m \\ -m \le k-l \le m \end{cases}} \text{flip}\left(\mathfrak{A}\right)(:,:,m+1-l)\mathfrak{B}(:,:,m+1-l+k) && (8)\\
&= \sum_{\begin{cases} 1 \le \sigma \le 2m+1 \\ 1 \le k+\sigma \le 2m+1 \end{cases}} \text{flip}\left(\mathfrak{A}\right)(:,:,\sigma)\mathfrak{B}(:,:,\sigma+k) && (9)\\
&= \sum_{\max(1,1-k) \le \sigma \le \min(2m+1,2m+1-k)} \text{flip}\left(\mathfrak{A}\right)(:,:,\sigma)\mathfrak{B}(:,:,\sigma+k) && (10)
\end{aligned}$$

Via matlab et tensorprod on peut aisément définir ce produit, le code suivant produit le produit de 2 3D array de phaseurs de dimensions compatibles, tronqué à des ordres différents, en considérant que tout phaseur au dela de la troncature est nul.

```
function D = hmq_times(A,B)
%Takes 2 3D array as input, seen as a Matrix sequence indexed on Z and
%truncated, stored along the third dimension.
%Compute the matrix convolution
%
% A is a 3D array of size m x n x mA
% B is a 3D array of size n x p x mB
%
    mA=(size(A,3)-1)/2;
    mB=(size(B,3)-1)/2;
    m=max(mB,mA);
    D=zeros(size(A,1),size(B,2),2*m+1);

    %Convolution using tensorprod
    for k=(-m:m)
        l1=max(k-mB,-mA);
        l2=min(k+mB,mA);
        V=A(:,:,(mA+1+l1):(mA+1+l2));
        U=B(:,:,(k+mB+1-l1):-1:(k+mB+1-l2));
        D(:,:,m+1+k)=tensorprod(V,U,[2,3],[1,3]);
    end

end
```

C. *Représentation des matrices Toeplitz*

On différencie 2 types de matrices associés aux phaseurs de $M(t)$:

- la matrice *Toeplitz par blocs* (anglais : block Toeplitz matrix), où des blocs sont répétés sur les diagonales
- la matrice en *Blocs-Toeplitzs* (anglais Toeplitz blocks matrix), qui est une matrice par bloc dont chacun des blocs est une matrice Toeplitz

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & \ddots & & \vdots \\ \vdots & & & \\ a_{m1} & & & a_{mn} \end{pmatrix} \tag{11}$$

Matrice A dont les coefficients dépendent du temps

$$\mathcal{T}_T^{\mathrm{c}}(A) = \left[\begin{array}{c|c|c|c} \mathcal{T}_T(a_{11}) & \mathcal{T}_T(a_{12}) & \cdots & \mathcal{T}_T(a_{1n}) \\ \hline \mathcal{T}_T(a_{21}) & \ddots & & \\ \hline \vdots & & & \\ \hline \mathcal{T}_T(a_{m1}) & & & \mathcal{T}_T(a_{mn}) \end{array} \right] \tag{12}$$

Représentation en blocs Toeplitzs

$$\mathcal{T}_T^{\mathrm{m}}(A) = \begin{pmatrix} \ddots & \ddots & \ddots & & \\ \ddots & A_0 & A_{-1} & A_{-2} & \\ \ddots & A_1 & A_0 & A_{-1} & \ddots \\ & A_2 & A_1 & A_0 & \ddots \\ & & \ddots & \ddots & \ddots \end{pmatrix} \tag{13}$$

Représentation Toeplitz par bloc