Constructing an orthonormal set of eigenvectors for DFT matrix using Gramians and determinants

Vadim Zaliva, lord@crocodile.org

July 17, 2012

Abstract

The problem of constructing an orthogonal set of eigenvectors for a DFT matrix is well studied. An elegant solution is mentioned by Matveev in [1]. In this paper, we present a distilled form of his solution including some steps unexplained in his paper, along with correction of typos and errors using more consistent notation. Then we compare the computational complexity of his method with the more traditional method involving direct application of the Gram-Schmidt process. Finally, we present our implementation of Matveev's method as a Mathematica module.

1 Definitions

The normalized matrix for discrete Fourier transform (DFT) of size n is defined as:

$$\Phi_{jk}(n) = \frac{1}{\sqrt{n}} w^{jk}, \quad j,k = 0,\ldots,n-1, \quad w = e^{i\frac{2\pi}{n}} \tag{1}$$

In some literature, an alternative definition is used where $w = e^{-i\frac{2\pi}{n}}$. It should be possible to adopt the algorithm described here with some minimal changes.

Throughout this paper, unless explicitly specified, we will use 0-based indices for matrices and arrays.

The scaling factor $\frac{1}{\sqrt{n}}$ ensures that Φ is unitary. An important for us property of a unitary matrix is that its eigenvectors corresponding to different eigenvalues are orthogonal[2].[1]

If e_k is an eigenvector of Φ with associated eigenvalue λ_k then by the definition of an eigenvector $\Phi e_k = \lambda_k e_k$. It is also a property of eigenvectors that $\Phi^q e_k = \lambda_k^q e_k$. In [3] it has been shown that $\Phi^4 = I$. This gives us $I e_k = \lambda^4 e_k$. From that, it follows that $\lambda_k^4 = 1$, and eigenvalues of DFT matrix are fourth roots of unity:

$$\lambda = (1, i, -1, -i) \tag{2}$$

The well known [1, 3] spectral decomposition of Φ into four orthogonal projections can be defined as:

---

[1]There is a typo in [2] stating that they are "orthonormal", while it should read "orthogonal" or "can be chosen orthonormal" instead. It has been reported to the author and acknowledged by him.

1

$$p_k = \frac{1}{4}\sum_{j=0}^{3}(-i)^{jk}\Phi^j, \quad k=0,\ldots,n-1 \tag{3}$$

Each projection matrix corresponds to one of four the possible eigenvalues from equation (2). The columns of each projection matrix are eigenvectors of Φ sharing the same eigenvalue.

As shown in [3], the multiplicity of the eigenvalue with a value of λ_k is equal to the trace of p_k. However, we can use simpler formulae from [1] to calculate the multiplicity of m_k:

$$
\begin{aligned}
m_o &= \lfloor \frac{n+1}{4} \rfloor, && \text{associated with } \lambda_0 = 1 \\
m_1 &= \lfloor \frac{n+2}{4} \rfloor, && \text{associated with } \lambda_1 = i \\
m_2 &= \lfloor \frac{n+3}{4} \rfloor - 1, && \text{associated with } \lambda_2 = -1 \\
m_3 &= \lfloor \frac{n}{4} \rfloor + 1, && \text{associated with } \lambda_3 = -i
\end{aligned} \tag{4}
$$

where $\lfloor \ldots \rfloor$ in the equation above denotes the *floor* function. Note that for convenience, λ_k is defined so that $\lambda_k = i^k$.

Finally, following [1], we define $v(m,k)$, where $m,k=0,\ldots,n-1$ as an n-dimensional vector which is equal to the m-th row (or m-th column due to matrix symmetry) of p_k:

$$v(m,k) = ([p_k]_{0,m}, [p_k]_{1,m}, \ldots, [p_k]_{n-1,m}) \tag{5}$$

In the formula above, $[p_k]_{m,n}$ denotes an element at row m and column n of projection matrix p_k. The p_k per equation (3) could be expanded as:

$$p_k = \frac{I + (-i)^k\Phi + (-i)^{2k}\Phi^2 + (-i)^{3k}\Phi^3}{4}$$

This allows us to write a formula, computing an element of p_k at position (j,m).

$$[p_k]_{j,m} = \frac{\delta_{j,m} + (-i)^k \frac{w^{jm}}{\sqrt{(n)}} + (-1)^k\delta_{(j+m \bmod n),0} + (-i)^{3k}\frac{w^{-jm}}{\sqrt{(n)}}}{4} \tag{6}$$

Using this, we can express the j-th element of a vector $v(m,k)$ from equation (5) as $v_j(m,k) = [p_k]_{j,m}$.

It should be noted that equation (6) differs slightly from the similar equation (22) in [1] accounting for a correction. The difference is in the arguments of the second Kronecker delta, representing Φ^2 which has the following form:

$$\Phi^2 = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 \end{pmatrix}$$

2

According to Matveev's formula, which incorrectly uses $\delta_{n-j,m}$, we get the incorrect result:

$$\delta_{n-j,m} = \begin{pmatrix} \mathbf{0} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 \end{pmatrix}$$

This is correct for all elements except the one at $(0,0)$, which should be 1 instead of 0. The expression $\delta_{(j+m \bmod n),0}$, which we are using instead, gives us the correct representation of Φ^2. An expression similar to ours is also used in [3].

2 Finding a complete system eigenvectors of $\Phi(n)$

Each projection matrix corresponds to one of four the possible eigenvalues from equation (2). The columns of each projection matrix are eigenvectors of Φ sharing the same eigenvalue. A complete set of eigenvectors of Φ spanning C^n could be constructed from columns of orthogonalized projection matrices taking the first m_k non-zero columns of p_k. The first column of p_1 and p_3 are formed by all zeros and have to be skipped. Thus, our set of n eigenvectors would consist of:

$$\begin{aligned} v(m,0), \quad & m = 0,1,\ldots m_0 - 1, \quad \text{associated with } \lambda_0 = 1 \\ v(m,1), \quad & m = 1,2,\ldots m_1, \quad \text{associated with } \lambda_1 = i \\ v(m,2), \quad & m = 0,1,\ldots m_2 - 1, \quad \text{associated with } \lambda_2 = -1 \\ v(m,3), \quad & m = 1,2,\ldots m_3, \quad \text{associated with } \lambda_3 = -i \end{aligned} \tag{7}$$

with $m_0 + m_1 + m_2 + m_3 = n$. The proof of this using Chebychev sets could be found in [4].

3 Orthonormalization

Eigenvectors corresponding to different eigenvalues are orthogonal. However, eigenvectors within the same projection matrix are not guaranteed to be orthogonal, so the associated set of eigenvectors does not possess the orthogonality property either.

A straightforward approach to get orthonormal eigenvectors as suggested in Candan[3] is to apply Gram-Schmidt process to all columns of each projection matrix. Each projection matrix p_k will have rank m_k and thus after normalization, the resulting orthonormalized vector set will contain exactly m_k non-zero vectors.

Matveev in [1] presents another approach to constructing an orthonormal basis based on the same principles as the Gram-Schmidt process but involving the use of Gramian matrices and determinants.

The calculations of the orthogonal basis of p_k involve m_k columns of p_k taken according to equation (7). We have two cases: one for odd values of $k = 1, 3$ and one for even values of $k = 0, 2$. Let us consider the case of even values first.

We can find a sequence of orthogonal vectors $(e_0(k), e_1(k), \ldots, e_{m_k-1}(k))$ spanning eigenspace p_k using Gramian matrices and determinants[5]:

$$e_0(k) = v(0,k),$$

$$\ldots$$

$$e_j(k) = \begin{vmatrix} \langle v(0,k), v(0,k) \rangle & \cdots & \langle v(0,k), v(j-1,k) \rangle & v(0,k) \\ \vdots & \vdots & \vdots & \vdots \\ \langle v(j-1,k), v(0,k) \rangle & \cdots & \langle v(j-1,k), v(j-1,k) \rangle & v(j-1,k) \\ \langle v(j,k), v(0,k) \rangle & \cdots & \langle v(j,k), v(j-1,k) \rangle & v(j,k) \end{vmatrix} \tag{8}$$

In the equation above, $\langle v, u \rangle$ denotes the inner product of the vectors v and u. The determinant notation assumes generic determinant formulation which is defined for matrices containing mixed scalar and vector entries. The determinant could be calculated using Laplace (cofactor) expansion.

It has been observed in [1] that p_k is in fact a Gramian matrix of a set of vectors $v(m,k), m = 0, \ldots, n-1$, such as $[p_k]_{j,m} = \langle v(j,k), v(m,k) \rangle$ Using this fact, we can replace $(j+1) \times j$ upper entries of the matrix in equation (8) with corresponding entries from p_k:

$$e_0(k) = v(0,k),$$

$$\ldots$$

$$e_j(k) = \begin{vmatrix} [p_k]_{0,0} & \cdots & [p_k]_{0,j-1} & v(0,k) \\ \vdots & \vdots & \vdots & \vdots \\ [p_k]_{j-1,0} & \cdots & [p_k]_{j-1,j-1} & v(j-1,k) \\ [p_k]_{j,0} & \cdots & [p_k]_{j,j-1} & v(j,k) \end{vmatrix} \tag{9}$$

The resulting system of vectors e_k is orthogonal but not yet orthonormal. Each vector is normalized by dividing by its norm. As shown in [5], the norm can be calculated by:

$$\|e_i\| = \sqrt{G_j G_{j+1}} \tag{10}$$

where G_j, G_{j+1} are principal minors of p_k of respective orders. They represent Gram determinants.

For $k = 1, 3$, we need to take into account the fact that the first row and the first column of $p_k, \quad k = 1, 3$ contain all zeros. Therefore, for these values of k, the equation (9) will become:

4

$$e_0(k) = v(1,k),$$

$$\dots$$

$$e_j(k) = \begin{vmatrix} [p_k]_{1,1} & \cdots & [p_k]_{1,j} & v(1,k) \\ \vdots & \vdots & \vdots & \vdots \\ [p_k]_{j,1} & \cdots & [p_k]_{j,j} & v(j,k) \\ [p_k]_{j+1,1} & \cdots & [p_k]_{j+1,j} & v(j+1,k) \end{vmatrix} \qquad (11)$$

and equation (10) will become:

$$\|e_i\| = \sqrt{\begin{vmatrix} [p_k]_{1,1} & \cdots & [p_k]_{1,j} \\ \vdots & \vdots & \vdots \\ [p_k]_{j,1} & \cdots & [p_k]_{j,j} \end{vmatrix} \begin{vmatrix} [p_k]_{1,1} & \cdots & [p_k]_{1,j+1} \\ \vdots & \vdots & \vdots \\ [p_k]_{j+1,1} & \cdots & [p_k]_{j+1,j+1} \end{vmatrix}} \qquad (12)$$

4 Computational complexity

Because computational complexity of Matveev's algorithm is very high, it is not very practical for large n. The complexity is mostly attributed to multiple cofactor expansion operations which have complexity of $O(n!)$.

For comparison: obtaining a set of non-orthogonal eigenvectors and orthonormalizing the set using the Modified Gram-Schmidt process would take just $2n^3$ floating point operations (FLOPS)[6] which translates to $O(n^3)$ in Big-O notation.

According to the equation (4) for any n, the multiplicities of different projections could differ at most by 2. That means for a reasonably big n, the dimensionality of each of the four eigenspaces of Φ is at approximately $\frac{n}{4}$.

Using this observation, the performance could be improved further by a factor of four by applying the Gram-Schmidt process to $m_k \approx \frac{n}{4}$ vectors from each projection, which gives us a total cost of $\frac{n^3}{4}$ FLOPS to orthogonalize the complete set of eigenvectors. Although an improvement, this is still $O(n^3)$.

5 Mathematica implementation

Listing 1 presents the full source code of the Mathematica module constructing a complete orthonormal set of eigenvectors of DFT matrix $\Phi(n)$. It performs all computations and returns the results in symbolic form. The code intention was to illustrate and validate the algorithm, and clarity and expressiveness were chosen over performance. It has been developed and tested with Mathematica version 8.

Since Mathematica's *Det* function does not work with matrices containing both scalars and vectors, we have implemented our own function *lDet* which finds the determinant of any matrix using cofactor expansion across rows. For the same reason, we must use our own function *rowMinor* instead of Mathematica's *Minors*.

Listing 1: Mathematica module source code

```
1  (* ::Package:: *)
2
3  BeginPackage["dfteigh'"];
4
5  dftEigen::usage = "Orthonormal_basis_of_DFT_matrix_of_rank_N";
6  lDet::usage = "Determinant_using_Laplace_expansion";
7  rowMinor::usage = "Row_minor";
8
9  Begin["'Private'"]
10
11 Clear[rowMinor]
12 rowMinor[m_,r_]:=Map[Rest,Delete[m,r]]
13
14 Clear[lDet]
15 lDet[m_] := Module[{rows},
16   rows = Length[m];
17   If[rows == 1, m[[1, 1]],
18    Sum[(-1)^(1+r)*m[[r, 1]]*lDet[rowMinor[m, r]],
19        {r,1,rows}]]
20 ]
21
22 Clear[dftEigen]
23 dftEigen[n_] := Module[{multiplicities, vj, e},
24
25   multiplicities = {Floor[n/4]+1, Floor[(n+1)/4],
26     Floor[(n+2)/4], Floor[(n+3)/4] - 1};
27
28   vj[k_, j_, m_] := (1/4)*(
29     KroneckerDelta[j, m] +
30     (-1)^k KroneckerDelta[Mod[j + m, n], 0] +
31     (-I)^k*Exp[(2*Pi*I*j*m)/n]/Sqrt[n]+
32     (-I)^(3*k) Exp[(2*Pi*I*(-j)*m)/n]/Sqrt[n]
33   );
34
35   e[j_,k_] := Module[{g,gv,z,xn,d0,d1},
36     z= If[OddQ[k],1,0];
37     If[j==0, Table[vj[k,y,z],{y,0,n-1}]/Sqrt[vj[k,0+z,0+z]],
38       g=Table[vj[k,y,x],{x,z, j+z},{y,z,j+z-1}];
39       gv=Table[{Table[vj[k,y,x],{y,0,n-1}]},{x,z,j+z}];
40       d0=Det[Table[vj[k,y,x],{x,z,j+z},{y,z,j+z}]];
41       d1=Det[Table[vj[k,y,x],{x,z,j+z-1},{y,z,j+z-1}]];
42       lDet[Join[g,gv,2]]/Sqrt[d0*d1]
43     ]
44   ];
45
46   Map[e[#[[1]],#[[2]]] &,
47   Flatten[Table[Transpose[{Range[0, multiplicities[[m+1]]-1],
48   ConstantArray[m, multiplicities[[m+1]]]}],{m,0,3}],1]]
```

6

```
49  ]
50
51  End [ ] ;
52  EndPackage [ ] ;
```

6   Examples

Using the Mathematica code above, we can calculate a set of eigenvectors for a DFT matrix of order 6. Combining them as columns of matrix O gives the following matrix, approximated numerically:

$$O_6 = \begin{pmatrix} 0.8391 & 0. & 0. & 0.5439 & 0. & 0. \\ 0.2433 & 0.5412 & 0.6533 & -0.3753 & 0.0843 & 0.2706 \\ 0.2433 & -0.2979 & 0.2706 & -0.3753 & -0.4596 & -0.6533 \\ 0.2433 & -0.4865 & 0. & -0.3753 & 0.7505 & 0. \\ 0.2433 & -0.2979 & -0.2706 & -0.3753 & -0.4596 & 0.6533 \\ 0.2433 & 0.5412 & -0.6533 & -0.3753 & 0.0843 & -0.2706 \end{pmatrix}$$

We can verify that it diagonalizes $\Phi(6)$ by calculating:

$$O_6^{-1}\Phi(6)O_6 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & i & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -i \end{pmatrix}$$

The diagonal contains eigenvalues repeating consistently with the associated multiplicities $m = (2, 1, 2, 1)$ and dimensions of the eigenspaces.

Taking the outer product of all columns of O_6 we can confirm that the set is indeed orthonormal:

$$\begin{pmatrix} 1. & 0. & 0. & 0. & 0. & 0. \\ 0. & 1. & 0. & 0. & 0. & 0. \\ 0. & 0. & 1. & 0. & 0. & 0. \\ 0. & 0. & 0. & 1. & 0. & 0. \\ 0. & 0. & 0. & 0. & 1. & 0. \\ 0. & 0. & 0. & 0. & 0. & 1. \end{pmatrix}$$

The Mathematica notebook, used to make calculations in the example above was:

Needs["dfteigh", ToFileName[NotebookDirectory[], "dfteigh.m"]];

$n = 6$;

$\Phi = \text{Table} \left[\frac{1}{\sqrt{n}}\text{Exp} \left[\frac{2\pi*i*k*m}{n} \right], \{k, 0, n-1\}, \{m, 0, n-1\} \right];$

eall = dftEigen[n];

$o = N$[Transpose[eall]]; MatrixForm[Round[o, 0.0001]];

MatrixForm[Round[o, 0.0001]]

MatrixForm $\left[\text{Round}\left[\text{Inverse}[o].N[\Phi].o, 10^{-10}\right]\right]$

MatrixForm[N[FullSimplify[Outer[Dot, eall, eall, 1]]]]

7 Acknowledgements

References

[1] Vladimir B Matveev. Interwining relations between the fourier transfom and discrete fourier transform, the related functional identities and beyond. *Inverse Problems*, 17:633, 2001.

[2] G. Strang. *Linear Algebra and Its Applications*. Thomson, Brooks/Cole, fourth edition, 2006.

[3] Ç. Candan. On the eigenstructure of DFT matrices [DSP education]. *Signal Processing Magazine, IEEE*, 28(2):105–108, 2011.

[4] J McClellan and T Parks. Eigenvalue and eigenvector decomposition of the discrete fourier transform, 1972.

[5] F R Gantmakher. *The Theory of Matrices*. Chelsea Publishing Co., 1959.

[6] G.H. Golub and C.F. Van Loan. *Matrix computations*, volume 3. Johns Hopkins Univ Pr, 1996.