Lecture 17: FFT and structured matrices

School of Mathematical Sciences, Xiamen University

1. Discrete Fourier transform and its inverse

Definition 1

The discrete Fourier transform (DFT) is a mapping on $\mathbb{C}^n$ given by

$$[\mathcal{F}_n\{\mathbf{f}\}]_i = \sum_{j=0}^{n-1} \omega_n^{ij} f_j, \quad i = 0, 1, \cdots, n-1,$$

where $\omega_n = \mathrm{e}^{-\mathrm{i}2\pi/n}$ and $\mathrm{i} = \sqrt{-1}$. The inverse DFT is given by

$$\left[\mathcal{F}_n^{-1}\{\mathbf{g}\}\right]_i = \frac{1}{n} \sum_{j=0}^{n-1} \omega_n^{-ij} g_j, \quad i = 0, 1, \cdots, n-1.$$

- DFT and inverse DFT as matrix-vector products:

$$\mathcal{F}_n\{\mathbf{f}\} = \mathbf{F}_n \mathbf{f}, \quad \mathcal{F}_n^{-1}\{\mathbf{g}\} = \frac{1}{n}\mathbf{F}_n^* \mathbf{g} = \frac{1}{n}\overline{\mathbf{F}_n \overline{\mathbf{g}}}, \quad \mathbf{F}_n = \left[\omega_n^{ij}\right]_{i,j=0}^{n-1}.$$

2. The FFT algorithm

- For simplicity, we assume that $n = 2^k$ and set $m = n/2$. Obviously,

$$\omega_m = \omega_n^2 = \mathrm{e}^{-\mathrm{i}2\pi/m}, \qquad \omega_m^m = 1, \qquad \omega_n^m = -1.$$

- Given any $\mathbf{f} = \begin{bmatrix} f_0 & f_1 & \cdots & f_{n-1} \end{bmatrix}^\top \in \mathbb{C}^n$, for $i = 0, 1, \ldots, m-1$,

$$\begin{aligned} [\mathcal{F}_n\{\mathbf{f}\}]_i &= \sum_{l=0}^{m-1} \omega_n^{i2l} f_{2l} + \sum_{l=0}^{m-1} \omega_n^{i(2l+1)} f_{2l+1} \\ &= \sum_{l=0}^{m-1} \omega_m^{il} f_{2l} + \omega_n^i \sum_{l=0}^{m-1} \omega_m^{il} f_{2l+1} \\ &= [\mathcal{F}_m\{\mathbf{f}_\mathrm{e}\}]_i + \omega_n^i [\mathcal{F}_m\{\mathbf{f}_\mathrm{o}\}]_i, \end{aligned}$$

where

$$\mathbf{f}_\mathrm{e} = \begin{bmatrix} f_0 & f_2 & \cdots & f_{n-2} \end{bmatrix}^\top, \quad \mathbf{f}_\mathrm{o} = \begin{bmatrix} f_1 & f_3 & \cdots & f_{n-1} \end{bmatrix}^\top.$$

- For $i = 0, 1, \ldots, m-1$, we also have

$$
\begin{aligned}
[\mathcal{F}_n\{\mathbf{f}\}]_{m+i} &= \sum_{l=0}^{m-1} \omega_n^{(m+i)2l} f_{2l} + \sum_{l=0}^{m-1} \omega_n^{(m+i)(2l+1)} f_{2l+1} \\
&= \sum_{l=0}^{m-1} \omega_m^{il} f_{2l} - \omega_n^i \sum_{l=0}^{m-1} \omega_m^{il} f_{2l+1} \\
&= [\mathcal{F}_m\{\mathbf{f}_{\mathrm{e}}\}]_i - \omega_n^i [\mathcal{F}_m\{\mathbf{f}_{\mathrm{o}}\}]_i.
\end{aligned}
$$

- Let $\mathrm{FFT}(n)$ denote the number of flops required to evaluate $\mathcal{F}_n\{\mathbf{f}\}$ by a recursive algorithm. Given the vectors $\mathcal{F}_m\{\mathbf{f}_{\mathrm{e}}\}$ and $\mathcal{F}_m\{\mathbf{f}_{\mathrm{o}}\}$, only m multiplications, m additions and m subtractions are needed to evaluate $\mathcal{F}_n\{\mathbf{f}\}$. Hence,

$$
\mathrm{FFT}(n) = 3m + 2\mathrm{FFT}(m) = 3n/2 + 2\mathrm{FFT}\left(n/2\right).
$$

Since $\mathrm{FFT}(1) = 0$, then

$$
\mathrm{FFT}(n) = 3n/2 \times k = \frac{3}{2} n \log n.
$$

3. Flop counts for frequently used algorithms

Method	Matrix ($m \geq n$)	Operation or Factorization	Flops
MV product	$\mathbf{A} \in \mathbb{C}^{n\times n}$	$\mathbf{b} = \mathbf{A}\mathbf{x}$	$2n^2$
FFT MV product	$\mathbf{F} \in \mathbb{C}^{n\times n}$	$\mathbf{b} = \mathbf{F}\mathbf{x}$	$3n \log n/2$
MM product	$\mathbf{A}, \mathbf{B} \in \mathbb{C}^{n\times n}$	$\mathbf{C} = \mathbf{A}\mathbf{B}$	$2n^3$
Inverse	$\mathbf{A} \in \mathbb{C}^{n\times n}$	$\mathbf{A}^{-1}$	$2n^3$
LU factorization	$\mathbf{A} \in \mathbb{C}^{n\times n}$	$\mathbf{P}\mathbf{A} = \mathbf{L}\mathbf{U}$	$2n^3/3$
Hessenberg LU	$\mathbf{H} \in \mathbb{C}^{n\times n}$	$\mathbf{H} = \mathbf{L}\mathbf{U}$	$2n^2$
Tridiagonal LU	$\mathbf{T} \in \mathbb{C}^{n\times n}$	$\mathbf{T} = \mathbf{L}\mathbf{U}$	$3n$
Cholesky	$\mathbf{A} \in \mathbb{C}^{n\times n}$	$\mathbf{A} = \mathbf{R}^*\mathbf{R}$	$n^3/3$
Triangular solve	$\mathbf{L} \in \mathbb{C}^{n\times n}$	$\mathbf{L}\mathbf{x} = \mathbf{b}$	n^2
Triangular inverse	$\mathbf{L} \in \mathbb{C}^{n\times n}$	$\mathbf{L}^{-1}$	$2n^3/3$
Normal equations	$\mathbf{A} \in \mathbb{C}^{m\times n}$	$\mathbf{A}^*\mathbf{A} = \mathbf{R}^*\mathbf{R}$	$mn^2 + n^3/3$
Householder QR	$\mathbf{A} \in \mathbb{C}^{m\times n}$	$\mathbf{Q}^*\mathbf{A} = \mathbf{R}$	$2(mn^2 - n^3/3)$
MGS QR	$\mathbf{A} \in \mathbb{C}^{m\times n}$	$\mathbf{A} = \mathbf{Q}_n\mathbf{R}_n$	$2mn^2$
Bidiagonalization	$\mathbf{A} \in \mathbb{C}^{m\times n}$	$\mathbf{B} = \mathbf{U}^*\mathbf{A}\mathbf{V}$	$4(mn^2 - n^3/3)$
Hessenberg reduction	$\mathbf{A} \in \mathbb{C}^{n\times n}$	$\mathbf{H} = \mathbf{Q}^*\mathbf{A}\mathbf{Q}$	$10n^3/3$
Tridiagonal reduction	$\mathbf{A} \in \mathbb{C}^{n\times n}$	$\mathbf{T} = \mathbf{Q}^*\mathbf{A}\mathbf{Q}$	$4n^3/3$

Remark 2

On modern computer architectures the communication costs in moving data between different levels of memory or between processors in a network can exceed the arithmetic costs by orders of magnitude.

4. Circulant matrix

Definition 3

An $n \times n$ matrix $\mathbf{C}$ is called circulant if it has the form

$$\mathbf{C} = \begin{bmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & \ddots & c_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ c_{n-2} & \ddots & c_1 & c_0 & c_{n-1} \\ c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{bmatrix}.$$

We indicate this situation by $\mathbf{C} = \mathbf{circ}(\mathbf{c})$, where

$$\mathbf{c} = \begin{bmatrix} c_0 & c_1 & \cdots & c_{n-1} \end{bmatrix}^\top \in \mathbb{C}^n$$

- Exercise: Generate a circulant matrix in Matlab.

Definition 4

The $n \times n$ circulant right shift matrix is given by

$$\mathbf{R} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix} = \mathbf{circ}\left(\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^\top \right).$$

- Obviously, if $\mathbf{C} = \mathbf{circ}(\mathbf{c})$, then $\mathbf{C} = \sum_{j=0}^{n-1} c_j \mathbf{R}^j$.

Lemma 5

Let $\omega_n = \mathrm{e}^{-\mathrm{i}2\pi/n}$. *Then*

$$\mathbf{R} = \frac{1}{n}\mathbf{F}_n^* \mathrm{diag}\{1, \omega_n, \omega_n^2, \cdots, \omega_n^{n-1}\}\mathbf{F}_n.$$

Theorem 6

If $\mathbf{C} = \mathbf{circ}(\mathbf{c})$, *then*

$$\mathbf{C} = \mathbf{F}_n^{-1}\mathrm{diag}\{\widehat{\mathbf{c}}\}\mathbf{F}_n = \frac{1}{n}\mathbf{F}_n^{*}\mathrm{diag}\{\widehat{\mathbf{c}}\}\mathbf{F}_n$$

where

$$\widehat{\mathbf{c}} = \mathbf{F}_n\mathbf{c}.$$

Fast algorithm 1: Circulant matrix-vector product $\mathbf{v} = \mathbf{Cu}$

Step 1: Compute $\widehat{\mathbf{c}} = \mathbf{F}_n\mathbf{c}$ and $\widehat{\mathbf{u}} = \mathbf{F}_n\mathbf{u}$ by FFT

Step 2: Compute the component-wise vector product $\widehat{\mathbf{v}} = \widehat{\mathbf{c}}.*\widehat{\mathbf{u}}$

Step 3: Compute $\mathbf{v} = \frac{1}{n}\mathbf{F}_n^{*}\widehat{\mathbf{v}}$ by iFFT

---

5. Toeplitz matrix

Definition 7

A matrix is called Toeplitz if it is constant along diagonals. An $n \times n$ Toeplitz matrix $\mathbf{T}$ has the form

$$\mathbf{T} = \begin{bmatrix} t_0 & t_{-1} & \cdots & t_{2-n} & t_{1-n} \\ t_1 & t_0 & t_{-1} & \ddots & t_{2-n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ t_{n-2} & \ddots & t_1 & t_0 & t_{-1} \\ t_{n-1} & t_{n-2} & \cdots & t_1 & t_0 \end{bmatrix}.$$

We indicate this situation by $\mathbf{T} = \mathbf{toep}(\mathbf{t})$, where

$$\mathbf{t} = \begin{bmatrix} t_{1-n} & \cdots & t_{-1} & t_0 & t_1 & \cdots & t_{n-1} \end{bmatrix}^\top \in \mathbb{C}^{2n-1}.$$

- Explore `toeplitz(c,r)` in Matlab.

- Define $\mathbf{S} = \mathbf{toep}(\mathbf{s})$, where

$$\mathbf{s} = \begin{bmatrix} t_1 & t_2 & \cdots & t_{n-1} & 0 & t_{1-n} & \cdots & t_{-2} & t_{-1} \end{bmatrix}^\top.$$

Then we have

$$\mathbf{T}^{\mathrm{ce}} := \begin{bmatrix} \mathbf{T} & \mathbf{S} \\ \mathbf{S} & \mathbf{T} \end{bmatrix} = \mathbf{circ}(\mathbf{t}^{\mathrm{ce}}),$$

where

$$\mathbf{t}^{\mathrm{ce}} = \begin{bmatrix} t_0 & t_1 & \cdots & t_{n-1} & 0 & t_{1-n} & \cdots & t_{-2} & t_{-1} \end{bmatrix}^\top \in \mathbb{C}^{2n}.$$

Note that

$$\begin{bmatrix} \mathbf{T} & \mathbf{S} \\ \mathbf{S} & \mathbf{T} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{Tu} \\ \mathbf{Su} \end{bmatrix}.$$

Using the fast algorithm for a circulant matrix-vector product, we obtain the following fast algorithm for a Toeplitz matrix-vector product $\mathbf{v} = \mathbf{Tu}$.

Fast algorithm 2: Toeplitz matrix-vector product $\mathbf{v} = \mathbf{T}\mathbf{u}$

Step 1: Compute $\widehat{\mathbf{t}^{\mathrm{ce}}} = \mathbf{F}_{2n}\mathbf{t}^{\mathrm{ce}}$ and $\widehat{\mathbf{u}^{\mathrm{ze}}} = \mathbf{F}_{2n}[\mathbf{u}^\top \ \mathbf{0}]^\top$ by FFT

Step 2: Compute the component-wise vector product $\widehat{\mathbf{w}} = \widehat{\mathbf{t}^{\mathrm{ce}}} .* \widehat{\mathbf{u}^{\mathrm{ze}}}$

Step 3: Compute $\mathbf{w} = \dfrac{1}{2n}\mathbf{F}_{2n}^*\widehat{\mathbf{w}}$ by iFFT

Step 4: Extract the first n components of $\mathbf{w}$ to obtain $\mathbf{v}$, i.e., $\mathbf{v} = \mathbf{w}(1:n)$

6. Hankel matrix

- A *Hankel* matrix $\mathbf{H} = \left[h_{ij}\right]$ has identical elements along all its anti-diagonals, meaning that

$$h_{ij} = h_{i+l,j-l}$$

for all relevant integers i, j, and l.

- Explore `hankel(c,r)` in Matlab.

- A Hankel matrix is symmetric by definition.
- The relation to a Toeplitz matrix: the matrix

$$\mathbf{T} = \mathbf{J}\mathbf{H}, \qquad \mathbf{J} = \begin{bmatrix} & & & 1 \\ & & 1 & \\ & \iddots & & \\ 1 & & & \end{bmatrix}$$

  is a Toeplitz matrix, where $\mathbf{J}$ is a permutation matrix obtained by reversing the columns (or rows) of the identity.
- Fast algorithm for a Hankel matrix-vector product can be obtained easily from that of a Toeplitz matrix-vector product.

7. Other issues

Discrete sine transform: `dst`

Discrete cosine transform: `dct`

Symmetric Toeplitz-plus-Hankel (STH) matrix ...

8. Kronecker product and vec(·) operator

Definition 8

Let $\mathbf{A} \in \mathbb{C}^{m\times n}$ and $\mathbf{B} \in \mathbb{C}^{p\times q}$. Then $\mathbf{A} \otimes \mathbf{B}$, the Kronecker product of $\mathbf{A}$ and $\mathbf{B}$, is the $mp \times nq$ matrix

$$\mathbf{A} \otimes \mathbf{B} := \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}.$$

Definition 9

Let $\mathbf{A} \in \mathbb{C}^{m\times n}$. Then $\mathrm{vec}(\mathbf{A})$ is defined to be a column vector of size mn made of the columns of $\mathbf{A}$ stacked atop one another from left to right.

- If $\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{bmatrix}$, then

$$\mathrm{vec}(\mathbf{A}) = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix}.$$

- Let $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{m \times n}$. Then $\mathrm{tr}(\mathbf{A}^*\mathbf{B}) = \mathrm{vec}(\mathbf{A})^*\mathrm{vec}(\mathbf{B})$.

Theorem 10

Let $\mathbf{A} \in \mathbb{C}^{p \times m}$, $\mathbf{X} \in \mathbb{C}^{m \times n}$, and $\mathbf{B} \in \mathbb{C}^{n \times q}$. Then the following properties hold:

$$\mathrm{vec}(\mathbf{AX}) = (\mathbf{I}_n \otimes \mathbf{A})\mathrm{vec}(\mathbf{X}),$$
$$\mathrm{vec}(\mathbf{XB}) = (\mathbf{B}^\top \otimes \mathbf{I}_m)\mathrm{vec}(\mathbf{X}),$$
$$\mathrm{vec}(\mathbf{AXB}) = (\mathbf{B}^\top \otimes \mathbf{A})\mathrm{vec}(\mathbf{X}).$$

Theorem 11

The following facts about Kronecker products hold:

$$\begin{aligned}
(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) &= (\mathbf{AC}) \otimes (\mathbf{BD}), \\
(\mathbf{A} \otimes \mathbf{B})^{-1} &= \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}, \\
(\mathbf{A} \otimes \mathbf{B})^{\dagger} &= \mathbf{A}^{\dagger} \otimes \mathbf{B}^{\dagger}, \\
(\mathbf{A} \otimes \mathbf{B})^{*} &= \mathbf{A}^{*} \otimes \mathbf{B}^{*}.
\end{aligned}$$

- Exercise: For $\mathbf{A} \in \mathbb{C}^{m\times n}$, $\mathbf{B} \in \mathbb{C}^{p\times q}$, and $\mathbf{C} \in \mathbb{C}^{m\times q}$, solve

$$\min_{\mathbf{X}\in\mathbb{C}^{n\times p}} \|\mathbf{AXB} - \mathbf{C}\|_{\mathrm{F}} = ?$$

- Exercise: Let $\mathcal{T}$ denote the triangular truncation operator, which is a linear operator that maps a given matrix to its strictly lower triangular part. Write down the matrix form of this operator.

- Exercise: Let $\mathbf{A} \in \mathbb{C}^{m\times m}$ and $\mathbf{B} \in \mathbb{C}^{n\times n}$. What are eigenvalues of

$$\mathbf{I} \otimes \mathbf{A} + \mathbf{B} \otimes \mathbf{I}, \quad \text{and} \quad \mathbf{A} \otimes \mathbf{B}?$$

9. Reference books for Toeplitz solvers and FFT

- Chan, Raymond Hon-Fu and Jin, Xiao-Qing
  An Introduction to Iterative Toeplitz Solvers, SIAM, 2007
- Van Loan, Charles
  Computational Frameworks for the Fast Fourier Transform, SIAM, 1992

10. Further reading for fast multipole methods

- Greengard, Leslie F. and Rokhlin, Vladimir V.
  A fast algorithm for particle simulations
  Journal of Computational Physics 72 (1987), 325-348.