LINEAR ALGEBRA AND VECTOR ANALYSIS

MATH 22B

Unit 33: DFT

Lecture

33.1. Remember the circular matrices like

$$Q = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} ?$$

You should. We have seen that they are orthogonal matrices for which we can find the eigenvalues explicitly. Indeed the characteristic polynomial $-\lambda^5 + 1$ has as roots the numbers $\lambda_k = e^{2\pi k i/5}$ with $k = 0, \cdots, 4$. What are the eigenvectors?

Problem A: Verify that if λ is an eigenvalue of Q, then $[1, \lambda, \lambda^2, \cdots \lambda^4]^T$ is an eigenvector of Q to the eigenvalue λ.

33.2. Remember that orthogonal matrices are normal. By the spectral theorem for normal matrices we know that Q can be diagonalized $B = S^*QS$, where S is unitary $S^*S = 1$ and B is diagonal with eigenvalues in the diagonal.

Problem B: Write down the unitary matrix S for the matrix A above.

33.3. Given $n = 5$ numbers a, b, c, d, e, we can look at the matrix $A = a + bQ + cQ^2 + dQ^3 + eQ^4$.

Problem C: How do you diagonalize the matrix A? What are the eigenvalues, what are the eigenvectors?

33.4. You have now already covered the discrete Fourier transform. The map, which assigns to the list $(a_1, a_2, a_3, a_4, a_5)$ the eigenvalues $\hat{a}(k) = \sum_j a_j e^{2\pi i jk/5}$ is called the **Discrete Fourier transform** abbreviated DFT. We can write

$$\hat{a} = Sa$$

where S is the unitary coordinate change matrix containing the eigenvectors as column vectors.

33.5. How can we use this? One application is the multiplication of numbers.

Problem D: Compute $32 * 45$ using school arithmetic.

33.6. Now form the matrices $A = 2 + 3Q$ and $B = 5 + 4Q$ and form $A * B$ to verify that $AB = 10 + 23Q + 12Q^2$.

$$AB = \begin{bmatrix} 2 & 3 & 0 & 0 & 0 \\ 0 & 2 & 3 & 0 & 0 \\ 0 & 0 & 2 & 3 & 0 \\ 0 & 0 & 0 & 2 & 3 \\ 3 & 0 & 0 & 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} 5 & 4 & 0 & 0 & 0 \\ 0 & 5 & 4 & 0 & 0 \\ 0 & 0 & 5 & 4 & 0 \\ 0 & 0 & 0 & 5 & 4 \\ 4 & 0 & 0 & 0 & 5 \end{bmatrix} = \begin{bmatrix} 10 & 23 & 12 & 0 & 0 \\ 0 & 10 & 23 & 12 & 0 \\ 0 & 0 & 10 & 23 & 12 \\ 12 & 0 & 0 & 10 & 23 \\ 23 & 12 & 0 & 0 & 10 \end{bmatrix} .$$

Problem D: What is $10 + 23 \cdot 10 + 12 \cdot 10^2$?

33.7. Well, this seems to be an awfully complicated way to compute the product of two numbers. Let us see what happened. We encoded the first number as a matrix A and then encoded the second matrix as a matrix B. After diagonalizing the two matrices, we just can compute their diagonal entries. This can be done fast. It turns out that the map from the number $a = (a_1, \dots, a_n)$ to its discrete Fourier transform $\hat{a} = (\hat{a}_1, \dots, \hat{a}n)$ can be computed fast.

Theorem: One can multiply two integers of length n in time $n \log(n)$

Sound

33.8. We can listen to a function by replacing Plot with Play:

```
Play[Sin[4 x] Abs[Sin[1000 x]], {x, 0, 3}]
```

Here is the function $f(x) = x$ periodically continued

```
Plot[SawtoothWave[x], {x, -Pi, Pi}]
```

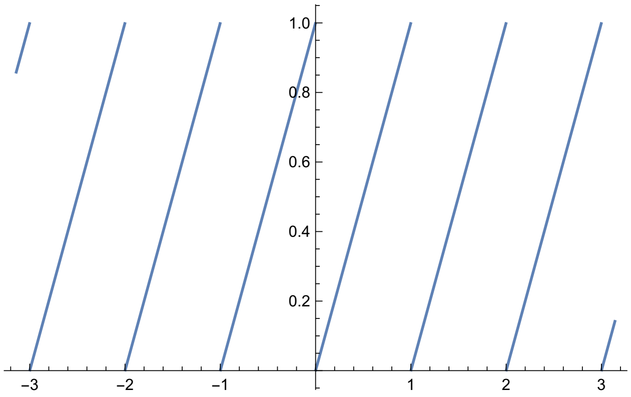

Figure 1. The Sawtooth Wave function.

```
Play[SawtoothWave[1000 x]], {x, 0, 1}]
```

It does not sound very nice. The reason is that the Fourier series of $f(x) = x$ does not decay very fast.

```
Play[Sin[Pi SawtoothWave[1000 x]], {x, -Pi, Pi}]
```

Problem F: Experiment with various functions and plot and play at least one.

33.9. If you have a sound file lying around, you can import it and look at the wave.

```
A=Import["https://www.quantumcalculus.org/sound/wiggle.wav"];
B=AudioData[A];
Audio[SampledSoundList[B, 48000]]
ListPlot[Table[B[[1, k]], {k, 1000}], Joined -> True]
```

33.10. Mathematica can play instruments. We can take the sound, represent the sound as a list of numbers and plot it:

```
A = Sound[SoundNote[1, 1, "Violin"]];
B = AudioData[A]; B1 = B[[1]]; B2 = B[[2]];
ListPlot[Table[B2[[k]], {k,500}], Joined -> True]
```

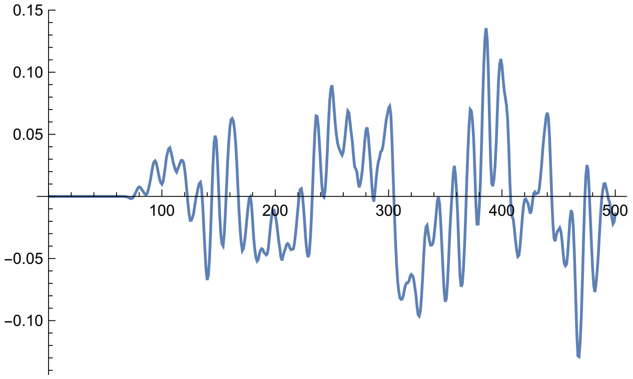

FIGURE 2. The first 500 entries in the sound wave representing a piano sample.

Homework

This homework is due on Tuesday, 4/23/2019.

Problem 33.1: The procedures Fourier and InverseFourier are implemented already in Mathematica Here are emulations showing what they do. Why does the $\sqrt{n}$ term appear in the matrix S?

```
X={5,9,7,4,2,3,2,1};
S=N[(Table[Exp[-I 2Pi k l/n],{k,0,n-1},{l,0,n-1}]/Sqrt[n])];
S.X
InverseFourier[X]
```

Now, let us get back X.

```
X={5,9,7,4,2,3,2,1};
S=N[(Table[Exp[I 2Pi k l/n],{k,0,n-1},{l,0,n-1}]/Sqrt[n])];
S.X
Fourier[X]
```

Problem 33.2: Make your own sound file (from a music piece or recording) and display part of it in Mathematica.

Problem 33.3: Mathematica has quite many MIDI instruments implemented. Use this to make your own little mini song.

Problem 33.4: Autotune is a technique which allows you to sing and never sing wrong. Explain how to implement this using Fourier theory.

Problem 33.5: Our ear can in the Cochlea do a Fourier analysis of sound. How does it work? Look it up and write a short paragraph explaining the principle.
Alternatively, if you should freak out thinking about your own ears, explain what the Mathematica code below does. Especially, why is “Reverse” used. Why is “PadRight” used?

```
FastMultiplication[x_,y_]:=Module[{X,Y,U,V,n,Q},
  X = Reverse[IntegerDigits[x]]; Y = Reverse[IntegerDigits[y]];
  n =Length[X]+Length[Y]+1; X=PadRight[X,n];     Y=PadRight[Y,n];
  U=InverseFourier[X]; V=InverseFourier[Y];
  Q=Round[Re[Fourier[U*V]*Sqrt[n]]];
  Sum[Q[[k]] 10^(k-1), {k,n}]];
x0 = 11234; y0 = 52342; FastMultiplication[x0,y0]==x0*y0
```