Contents

1 Discrete Fourier Transform – What is it? 2
1.1 General Interpolation Problem . . . 2
1.2 Discrete Fourier Transform . . . 3
1.3 Solving system of equations . . . 4
1.4 What if initial data is all real? . . . 5

2 Solve the linear system fast 6

3 DFT – What is it good for? 7
3.1 Convolution & Polynomial Multiplication . . . 7
3.2 Fast Polynomial Multiplication . . . 8
3.3 Convolution . . . 9
3.4 Sample Basis Functions . . . 10
3.5 Filtering out Noise in a Noise Signal . . . 13

1 Discrete Fourier Transform – What is it?

1.1 General Interpolation Problem

Given a set of data $y_0, \ldots, y_{2N-1}$, we would like to interpolate it with the function

$$\phi(x) = \sum_{n=0}^{2N-1} c_n \phi_n(x)$$

satisfying the conditions

$$\phi(x_m) = y_m, \text{ for } m = 0, \ldots, 2N-1.$$

The result is a linear system

$$\begin{aligned} \sum_{n=0}^{2N-1} c_n \phi_n(x_0) &= y_0 \\ \sum_{n=0}^{2N-1} c_n \phi_n(x_1) &= y_1 \\ &\vdots \\ \sum_{n=0}^{2N-1} c_n \phi_n(x_{2N-1}) &= y_{2N-1} \end{aligned}$$

or in matrix form $\mathbf{\Phi c} = \mathbf{y}$:

$$\begin{pmatrix} \phi_0(x_0) & \phi_1(x_0) & \cdots & \phi_{2N-1}(x_0) \\ \phi_0(x_1) & \phi_1(x_1) & \cdots & \phi_{2N-1}(x_1) \\ \vdots & \vdots & & \vdots \\ \phi_0(x_{2N-1}) & \phi_1(x_{2N-1}) & \cdots & \phi_{2N-1}(x_{2N-1}) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{2N-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{2N-1} \end{pmatrix},$$

where the m, n-th entry in the matrix $\mathbf{\Phi}$ is $\phi_n(x_m)$ We will see later that we can write $\mathbf{c} = \mathbf{Fy}$, with $\mathbf{F} = \mathbf{\Phi}^{-1}$ has the same form as $\mathbf{\Phi}$ (up to scaling).

1.2 Discrete Fourier Transform

We make a particular choice for the knots x_m and functions ϕ_n, for $m, n = 0, \ldots, 2N-1$. We use knots

$$x_m = e^{2\pi m i/2N} = \cos\left(\frac{2\pi m}{2N}\right) + i\sin\left(\frac{2\pi m}{2N}\right), \text{ for } m = 0, \ldots, 2N-1,$$

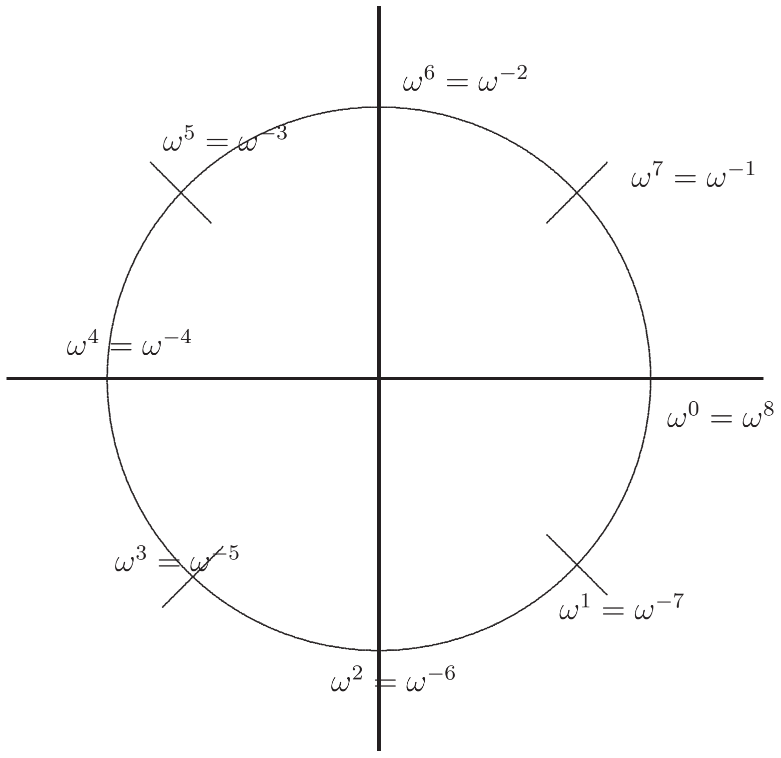
where $i = \sqrt{-1}$. Letting $\omega = e^{-2\pi i/2N}$ denote the $2N$-th root of 1 in the complex plane, we note that $x_m = \omega^{-m} = \omega^{2N-m}$, where $\omega^{2N} = 1$.

For functions ϕ_n we choose $\phi_n(x) = \frac{1}{2N}x^n$, for $n = 0, \ldots, 2N$. We choose this scaling to match that used in the text. Other than the scaling and the choice of knots, this looks like ordinary polynomial interpolation.

For example, when $2N = 8$, the system of equations becomes

$$\frac{1}{8}\begin{pmatrix} \omega^{-0} & \omega^{-0} & \omega^{-0} & \omega^{-0} & \omega^{-0} & \omega^{-0} & \omega^{-0} & \omega^{-0} \\ \omega^{-0} & \omega^{-1} & \omega^{-2} & \omega^{-3} & \omega^{-4} & \omega^{-5} & \omega^{-6} & \omega^{-7} \\ \omega^{-0} & \omega^{-2} & \omega^{-4} & \omega^{-6} & \omega^{-0} & \omega^{-2} & \omega^{-4} & \omega^{-6} \\ \omega^{-0} & \omega^{-3} & \omega^{-6} & \omega^{-1} & \omega^{-4} & \omega^{-7} & \omega^{-2} & \omega^{-5} \\ \omega^{-0} & \omega^{-4} & \omega^{-0} & \omega^{-4} & \omega^{-0} & \omega^{-4} & \omega^{-0} & \omega^{-4} \\ \omega^{-0} & \omega^{-5} & \omega^{-2} & \omega^{-7} & \omega^{-4} & \omega^{-1} & \omega^{-6} & \omega^{-3} \\ \omega^{-0} & \omega^{-6} & \omega^{-4} & \omega^{-2} & \omega^{-0} & \omega^{-6} & \omega^{-4} & \omega^{-2} \\ \omega^{-0} & \omega^{-7} & \omega^{-6} & \omega^{-5} & \omega^{-4} & \omega^{-3} & \omega^{-2} & \omega^{-1} \end{pmatrix}\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} \tag{1}$$

where the values ω^n lie equally spaced around the unit circle in the complex plane:



$$\begin{aligned} \omega^0 &= \omega^8 &&= 1 \\ \omega^1 &= \omega^{-7} &&= \tfrac{1-i}{\sqrt{2}} \\ \omega^2 &= \omega^{-6} &&= -i \\ \omega^3 &= \omega^{-5} &&= \tfrac{-1-i}{\sqrt{2}} \\ \omega^4 &= \omega^{-4} &&= -1 \\ \omega^5 &= \omega^{-3} &&= \tfrac{-1+i}{\sqrt{2}} \\ \omega^6 &= \omega^{-2} &&= +i \\ \omega^7 &= \omega^{-1} &&= \tfrac{1+i}{\sqrt{2}} \end{aligned}$$

1.3 Solving system of equations

Normally the solution of a $2N \times 2N$ system of equations requires some sort of elimination process, but in this case the matrix involved has the special property that its conjugate transpose is almost its inverse. Hence we can compute the DFT of the sequence $\mathbf{y} = [y_0, y_1, \ldots]$ by multiplying by a matrix of the same special form as in equ. (1). For example, for $2N = 8$, the DFT of $\mathbf{y}$ is

$$\mathbf{c} = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{pmatrix} = \mathrm{DFT}(\mathbf{y}) = \begin{pmatrix} \omega^0 & \omega^0 & \omega^0 & \omega^0 & \omega^0 & \omega^0 & \omega^0 & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\ \omega^0 & \omega^2 & \omega^4 & \omega^6 & \omega^0 & \omega^2 & \omega^4 & \omega^6 \\ \omega^0 & \omega^3 & \omega^6 & \omega^1 & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\ \omega^0 & \omega^4 & \omega^0 & \omega^4 & \omega^0 & \omega^4 & \omega^0 & \omega^4 \\ \omega^0 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega^1 & \omega^6 & \omega^3 \\ \omega^0 & \omega^6 & \omega^4 & \omega^2 & \omega^0 & \omega^6 & \omega^4 & \omega^2 \\ \omega^0 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega^1 \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} \tag{2}$$

If we denote the matrix in the above equation (2) as $\mathbf{F}_8$, the equations can be written can be written as

$$\mathbf{c} = \mathrm{DFT}(\mathbf{y}) = \mathbf{F}_{2N}\mathbf{y} \quad \text{and} \quad \mathbf{y} = \mathrm{IDFT}(\mathbf{c}) = \frac{1}{2N}\overline{\mathbf{F}}_{2N}\mathbf{c},$$

where $\overline{\Box}$ denotes the complex conjugate of $\Box$ (elementwise if $\Box$ is a matrix or vector).

1.4 What if initial data is all real?

If all the y's are real, then the coefficients satisfy special properties. For example, in (2), rows 0 and 4 are real, and the other rows are in complex conjugate pairs: $1 \leftrightarrow 7$, $2 \leftrightarrow 6$, $3 \leftrightarrow 5$. Hence the corresponding entries in the DFT vector $\mathbf{c}$ will share these properties.

Formally, write the function $\phi(x)$ as

$$\begin{aligned}
\phi(x) &= c_0\phi_0(x) + \sum_{n=1}^{N-1} c_n\phi_n(x) + c_N\phi_N(x) + \sum_{n=N+1}^{2N-1} c_n\phi_n(x)) \\
&= c_0\phi_0(x) + \sum_{n=1}^{N-1} c_n\phi_n(x) + c_N\phi_N(x) + \sum_{n=1}^{N-1} c_{2N-n}\phi_{2N-n}(x)) \\
&= c_0\phi_0(x) + \sum_{n=1}^{N-1} [c_n\phi_n(x) + c_{2N-n}\phi_{2N-n}(x)] + c_N\phi_N(x) \\
&= \left(c_0 + \sum_{n=1}^{N-1} [c_n x^n + c_{2N-n}x^{2N-n}] + c_N x^N\right)/(2N).
\end{aligned}$$

At the knot $x_m = \omega^{-m}$, we have the relation $x_m^{-n} = x_m^{2N-n} = \overline{x_m^n}$ (the complex conjugate of x_m^n), so at the knots, we can write the above

$$\begin{aligned}
\phi(x_m) &= \left(c_0 + \sum_{n=1}^{N-1} (c_n x_m^n + c_{2N-n}x_m^{-n}) + c_N x_m^N\right)/(2N) \\
&= \left(c_0 + \sum_{n=1}^{N-1} (c_n x_m^n + c_{2N-n}\overline{x_m^n}) + c_N x_m^N\right)/(2N).
\end{aligned}$$

Also notice that $x_N = \omega^{-N} = \pm 1$ is real.

So, in order for $\phi(x)$ to be real at all the knots, the coefficients c_0 and c_N must be real, and the the remaining coefficients must satisfy $c_{2N-n} = \bar{c}_n$, for $n = 1, \dots, N-1$.

Under these conditions, we can split c_n into its real and imaginary parts: $c_n = a_n + ib_n$ and use the identities (with $c = a + ib$, $x = \cos\psi + i\sin\psi$, for some angle ψ)

$$\begin{aligned}
cx + \bar{c}\bar{x} &= (a+ib)(\cos\psi + i\sin\psi) + (a-ib)(\cos\psi - i\sin\psi) \\
&= 2a\cos\psi - 2b\sin\psi
\end{aligned}$$

and

$$x_m^n = (e^{im\psi})^n = \cos nm\psi + i\sin nm\psi$$

to write $\phi(x_m)$ as a sum of all real entries:

$$\boxed{\begin{aligned}
\phi(x_m) &= \left(a_0 + a_N x_m^N + \sum_{n=1}^{N-1} (a_n + ib_n)x_m^n + (a_n - ib_n)\overline{x_m^n}\right)/(2N) \\
&= \left(a_0 + a_N\cos(Nm\psi) + 2\sum_{n=1}^{N-1} a_n\cos(nm\psi) - b_n\sin(nm\psi)\right)/(2N).
\end{aligned}}$$

2 Solve the linear system fast

First note that $\mathbf{F}^H\mathbf{F} = 2N\mathbf{I}$, where $\mathbf{I}$ denotes the identity matrix, and $\mathbf{M}^H$ denotes the *conjugate transpose* of $\mathbf{M}$. So solving the linear system $(1/(2N)\overline{\mathbf{F}}\mathbf{c} = \mathbf{y}$ is as easy as multiplying by $\mathbf{F}$.

Second note that we can decompose $\mathbf{F}$ as follows. Since $\mathbf{F}$ is $2N \times 2N$, let us denote this matrix as $\mathbf{F}_{2N}$, and denote by $\mathbf{F}_N$ the matrix obtained by solving the half-sized problem. Let $\mathbf{D} = \textsc{diag}\{1, \omega, \omega^2, \ldots, \omega^{N-1}\}$. Then

$$\begin{pmatrix} \mathbf{F}_N & \mathbf{D}\mathbf{F}_N \\ \mathbf{F}_N & -\mathbf{D}\mathbf{F}_N \end{pmatrix} = \mathbf{F}_{2N} \cdot \text{column\_shuffle}$$

is a column permutation of $\mathbf{F}_{2N}$ (specifically the reverse of a perfect shuffle):

$$\begin{array}{rcl} [0,1,2,\ldots,N{-}1,N,\ldots,2N{-}1] & \mapsto & [0,2,4,\ldots,2N{-}2,1,3,4,\ldots,2N{-}1] \\ [0,N,1,N{+}1,2,N{+}2,\ldots,N{-}1,2N{-}1] & \leftarrow\!\shortmid & [0,1,2,\ldots,N{-}1,N,\ldots,2N{-}1] \end{array}$$

For example, for the DFT of a vector with $2N = 8$ elements, we have the half-sized mapping:

$$\mathbf{F}_4 = \begin{pmatrix} \omega^0 & \omega^0 & \omega^0 & \omega^0 \\ \omega^0 & \omega^2 & \omega^4 & \omega^6 \\ \omega^0 & \omega^4 & \omega^0 & \omega^4 \\ \omega^0 & \omega^6 & \omega^4 & \omega^2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix}$$

and the diagonal scaling matrix:

$$\mathbf{D} = \begin{pmatrix} \omega^0 & 0 & 0 & 0 \\ 0 & \omega^1 & 0 & 0 \\ 0 & 0 & \omega^2 & 0 \\ 0 & 0 & 0 & \omega^3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1-i}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & -i & 0 \\ 0 & 0 & 0 & \frac{-1-i}{\sqrt{2}} \end{pmatrix}.$$

Then the full-sized $\mathbf{F}_8$ can be written in terms of the half-sized quantities:

$$\begin{pmatrix} \mathbf{F}_4 & \mathbf{D}\mathbf{F}_4 \\ \mathbf{F}_4 & -\mathbf{D}\mathbf{F}_4 \end{pmatrix} = \left(\begin{array}{cccc|cccc} \omega^0 & \omega^0 & \omega^0 & \omega^0 & \omega^0 & \omega^0 & \omega^0 & \omega^0 \\ \omega^0 & \omega^2 & \omega^4 & \omega^6 & \omega^1 & \omega^3 & \omega^5 & \omega^7 \\ \omega^0 & \omega^4 & \omega^0 & \omega^4 & \omega^2 & \omega^6 & \omega^2 & \omega^6 \\ \omega^0 & \omega^6 & \omega^4 & \omega^2 & \omega^3 & \omega^1 & \omega^7 & \omega^5 \\ \hline \omega^0 & \omega^0 & \omega^0 & \omega^0 & \omega^4 & \omega^4 & \omega^4 & \omega^4 \\ \omega^0 & \omega^2 & \omega^4 & \omega^6 & \omega^5 & \omega^7 & \omega^1 & \omega^5 \\ \omega^0 & \omega^4 & \omega^0 & \omega^4 & \omega^6 & \omega^2 & \omega^6 & \omega^2 \\ \omega^0 & \omega^6 & \omega^4 & \omega^2 & \omega^7 & \omega^5 & \omega^3 & \omega^1 \end{array}\right) = \mathbf{F}_8 \cdot \text{column\_shuffle}.$$

This means that multiplying a vector by $\mathbf{F}_8$ can be reduced to a perfect shuffle of the entries, two multiplications by $\mathbf{F}_4$, and a diagonal scaling by $\mathbf{D}$ (4 multiplications). In general multiplying by $\mathbf{F}_{2N}$ can be reduced to 2 multiplications by $\mathbf{F}_N$ plus N multiplies in the diagonal scaling. If $S(N)$ is the cost of doing this for the $N \times N$ case, then we have the recurrence for the cost (assuming $N = 2^k$):

$$S(2^{k+1}) = 2S(2^k) + N$$

whose solution is

$$S(N) = S(2^k) = \frac{1}{2} \cdot 2^k \cdot k = \frac{1}{2} \cdot N \cdot \log_2 N.$$

3 DFT – What is it good for?

3.1 Convolution & Polynomial Multiplication

Let $A = [a_0, a_1, \ldots, a_n]$ and $B = [b_0, b_1, \ldots, b_p]$ be two signals. The convolution $A * B$ is

$$[a_0b_0,\ a_0b_1{+}a_1b_0,\ a_0b_2{+}a_1b_1{+}a_2b_0,\ \ldots\ a_mb_p].$$

Example: B is a smoothing filter: $B = [1, 2, 1]$. Then $A * B =$

$$[a_0, 2a_0{+}a_1, a_0{+}2a_1{+}a_2, a_1{+}2a_2{+}a_3, \ldots a_{m-2}{+}2a_{m-1}{+}a_m, a_{m-1}{+}2a_m, a_m].$$

Example:

$$A = [1, 2, 3, 1, 2, 3, 1, 2, 3] \quad B = [1, 2, 1]$$

yields

$$A * B = [1, 4, 8, 9, 7, 8, 9, 7, 8, 8, 3]$$

(relatively smoother).

Let $a(x) = a_0{+}a_1x{+}a_2x^2{+}\cdots{+}a_mx^m$ and $b(x) = b_0{+}b_1x{+}b_2x^2{+}\cdots{+}b_px^p$. Then the coefficients of $a(x) \cdot b(x)$ are just the entries of $A * B$.

3.2 Fast Polynomial Multiplication

The product $a(x) \cdot b(x)$ has degree $n{+}p$ so it is completely determined by its values at the $n{+}p{+}1$ knots $x_0, \ldots, x_{n+p+1}$. Choose the knots to be the $n{+}p{+}1$-th roots of unity on the unit circle. Then the vector of values of $a(x_k)$ would be just the DFT($[a_0, \ldots, a_n, 0, \ldots, 0]$), and the vector of values $b(x_k)$ would be just the DFT($[b_0, \ldots, b_p, 0, \ldots, 0]$). So we get

$$\mathrm{DFT}\begin{pmatrix}1\\4\\8\\9\\7\\8\\9\\7\\8\\8\\3\end{pmatrix} \cdot \frac{1}{4} = \begin{pmatrix}18.00\\ \text{-3.59 -.3121i}\\ \text{-2.64 -.2921i}\\ \text{-1.91+.168i}\\ \text{.5173 -.5515i}\\ \text{1.E-3 -.0119i}\\ \text{1.E-3+.011i}\\ \text{.5173+.551i}\\ \text{-1.91 -.1688i}\\ \text{-2.64+.292i}\\ \text{-3.59+.312i}\end{pmatrix} = \begin{pmatrix}18.0\\ \text{-3.0 -2.39i}\\ \text{-1.1 -3.56i}\\ \text{.244 -4.46i}\\ \text{.452-4.3i}\\ \text{.077-.59i}\\ \text{.077 -.59i}\\ \text{.452 -4.35i}\\ \text{.244-4.4i}\\ \text{-1.1-3.5i}\\ \text{-3.0-2.3i}\end{pmatrix} \odot \begin{pmatrix}1.000\\ \text{.7744 -.4977i}\\ \text{.2939 -.6437i}\\ \text{-.061 -.4244i}\\ \text{-.113 -.1304i}\\ \text{-.019 -5.E-3i}\\ \text{-.019+5E-3i}\\ \text{-.113+.130i}\\ \text{-.061+.424i}\\ \text{.2939+.643i}\\ \text{.7744+.497i}\end{pmatrix} = \mathrm{DFT}\left\{\begin{pmatrix}1\\2\\3\\1\\2\\3\\1\\2\\3\\0\\0\end{pmatrix} * \begin{pmatrix}1\\2\\1\\0\\0\\0\\0\\0\\0\\0\\0\end{pmatrix} \cdot \frac{1}{4}\right\} \quad (3)$$

This is much faster with the DFT, $O(n{+}p)\log(n{+}p)$ instead of $O(np)$. This example is illustrated in Fig. 1.

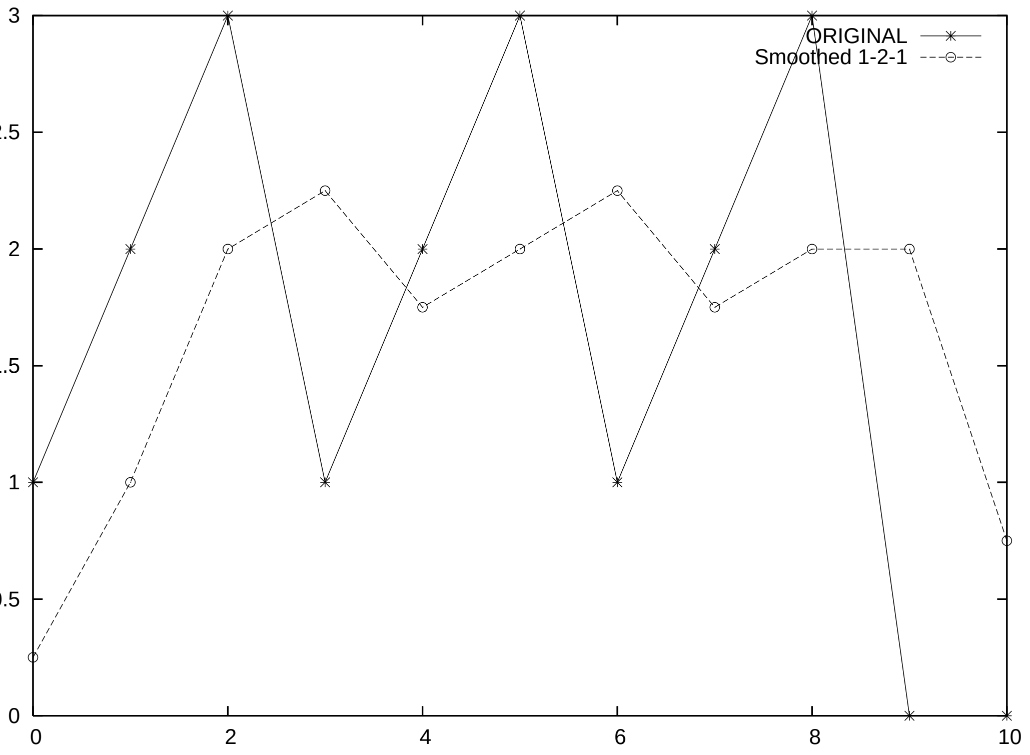


Figure 1: DFT example: diagram of convolution in equation (3), showing how the 1-2-1 kernel acts as a smoother.

3.3 Convolution

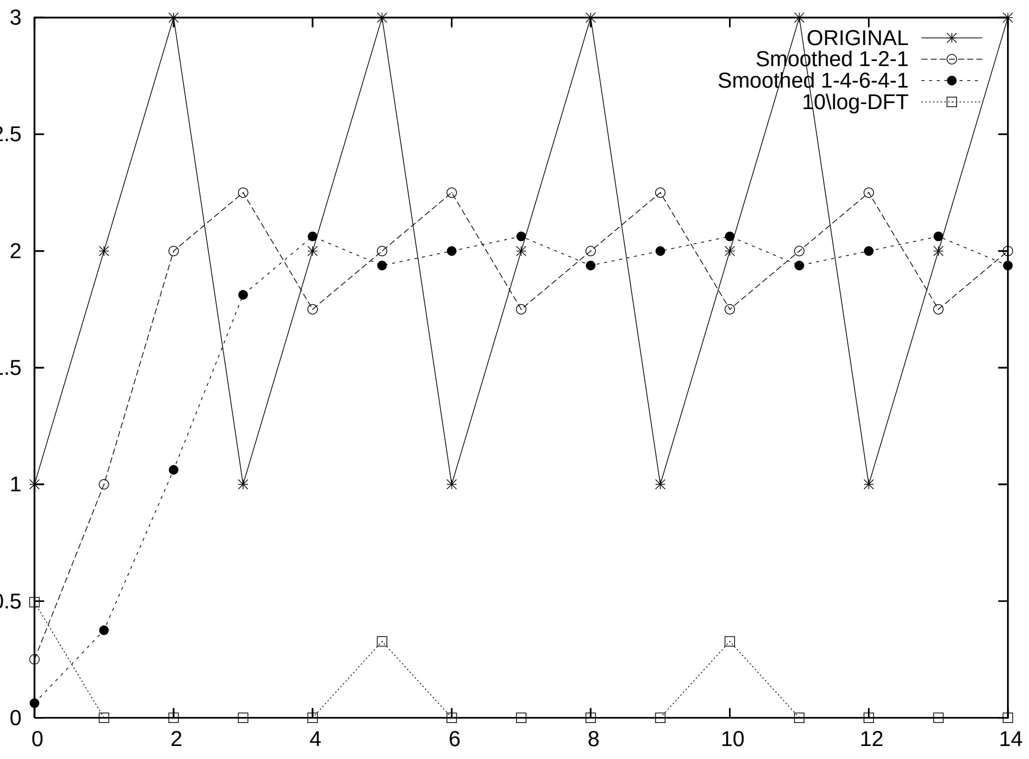


Figure 2: DFT example: diagram of convolution of a longer sequence, showing how the 1-2-1 kernel acts as a smoother, how 1-4-6-4-1 acts as an even bogger smoother. The curve along the bottom shows the magnitude of the DFT coefficients: the only ones that are non-zero are the coefficients in positions 0, 5, 10. Since there are $N = 15$ coefficients, position 10 is simply the reflection of position 5: $10 = N - 5$, so this shows the presence of only one frequency in the original sequence, plus a DC component.

3.4 Sample Basis Functions

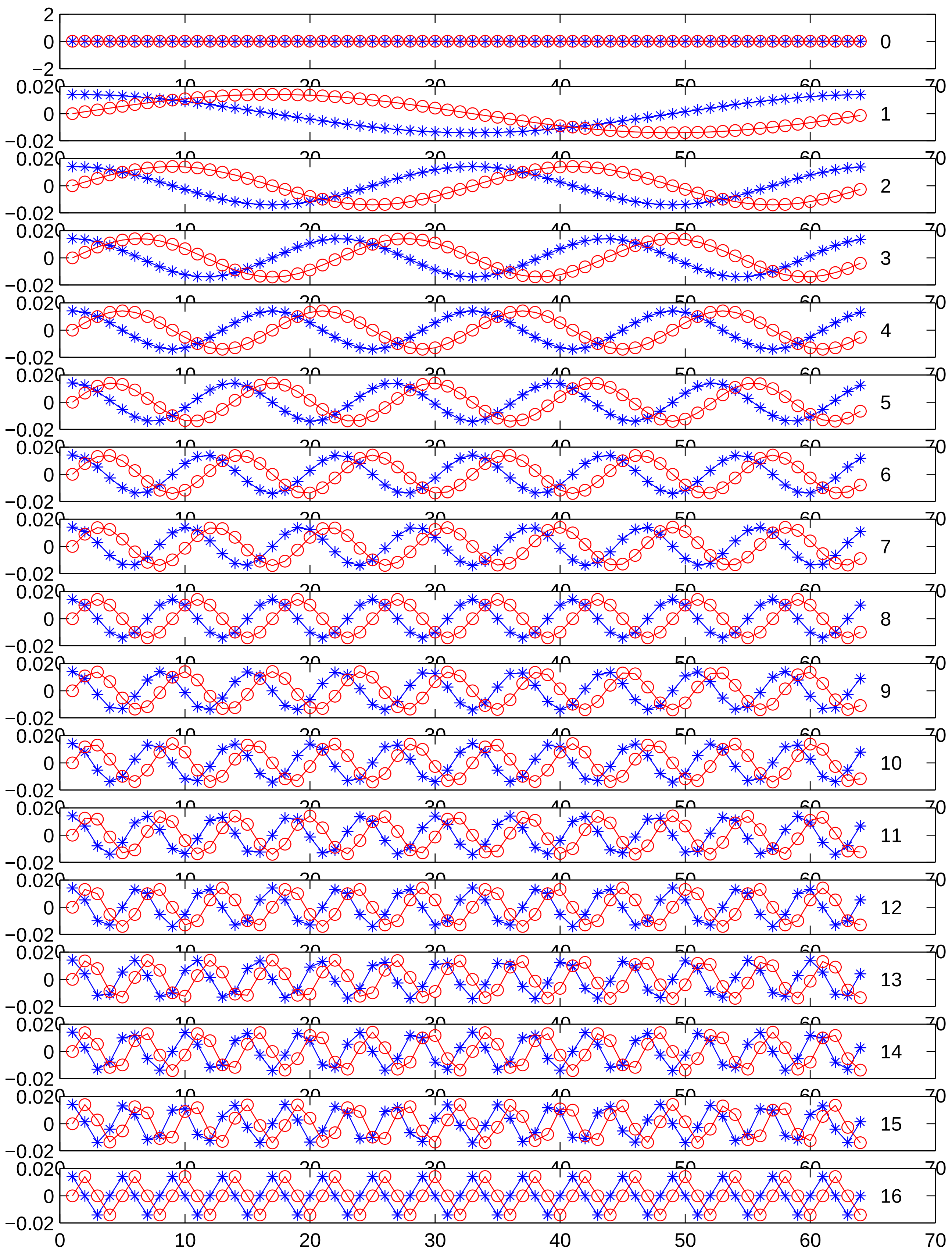


Figure 3: Sample Basis Functions: cosines in blue stars, sines in red circles

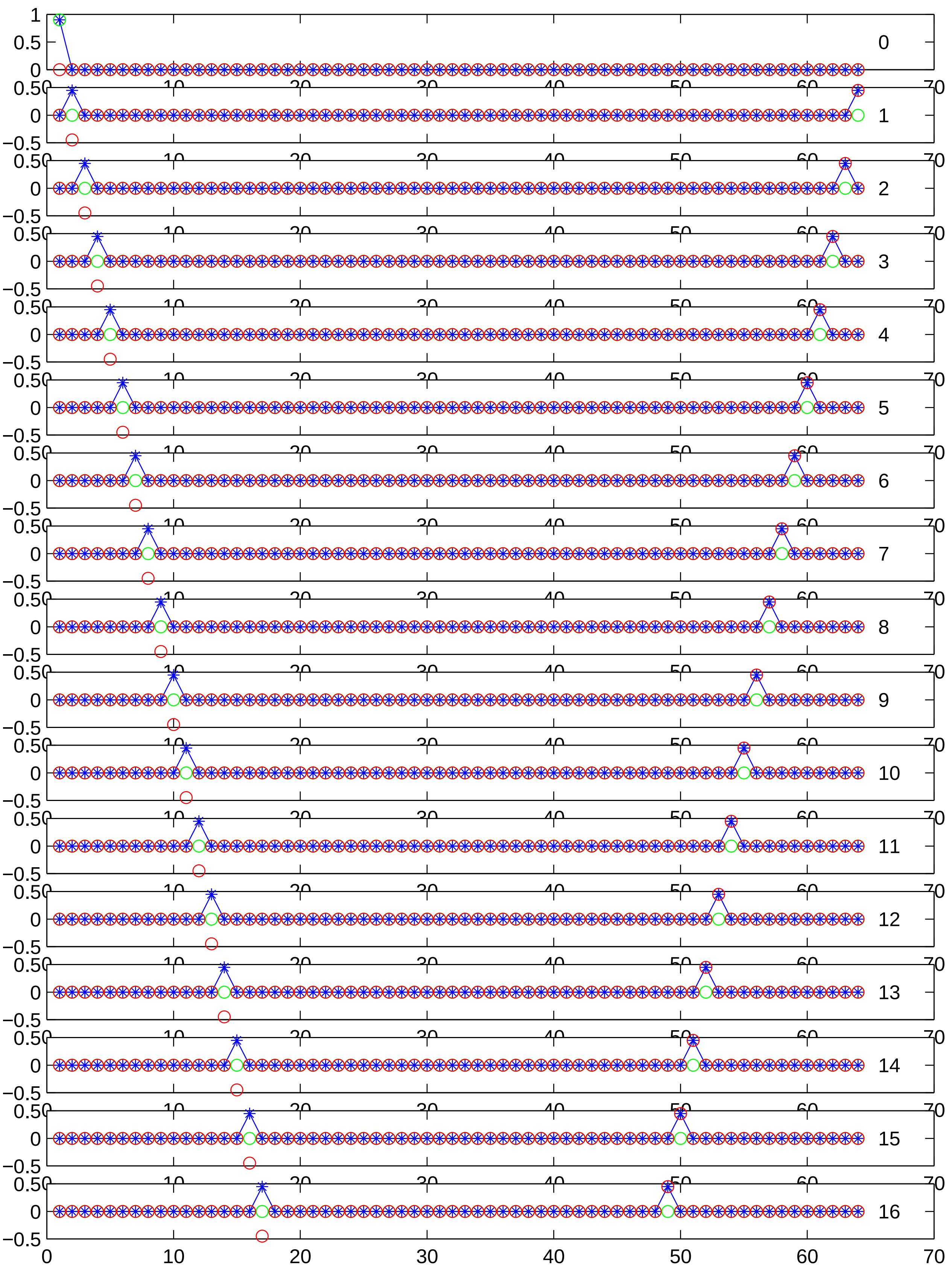


Figure 4: Discrete Fourier Transform (DFT) of Sample Basis Functions: transform of cosines in blue (all real), transform of sines in green (real part, all zero) and red (imaginary part).

3.5 Filtering out Noise in a Noise Signal

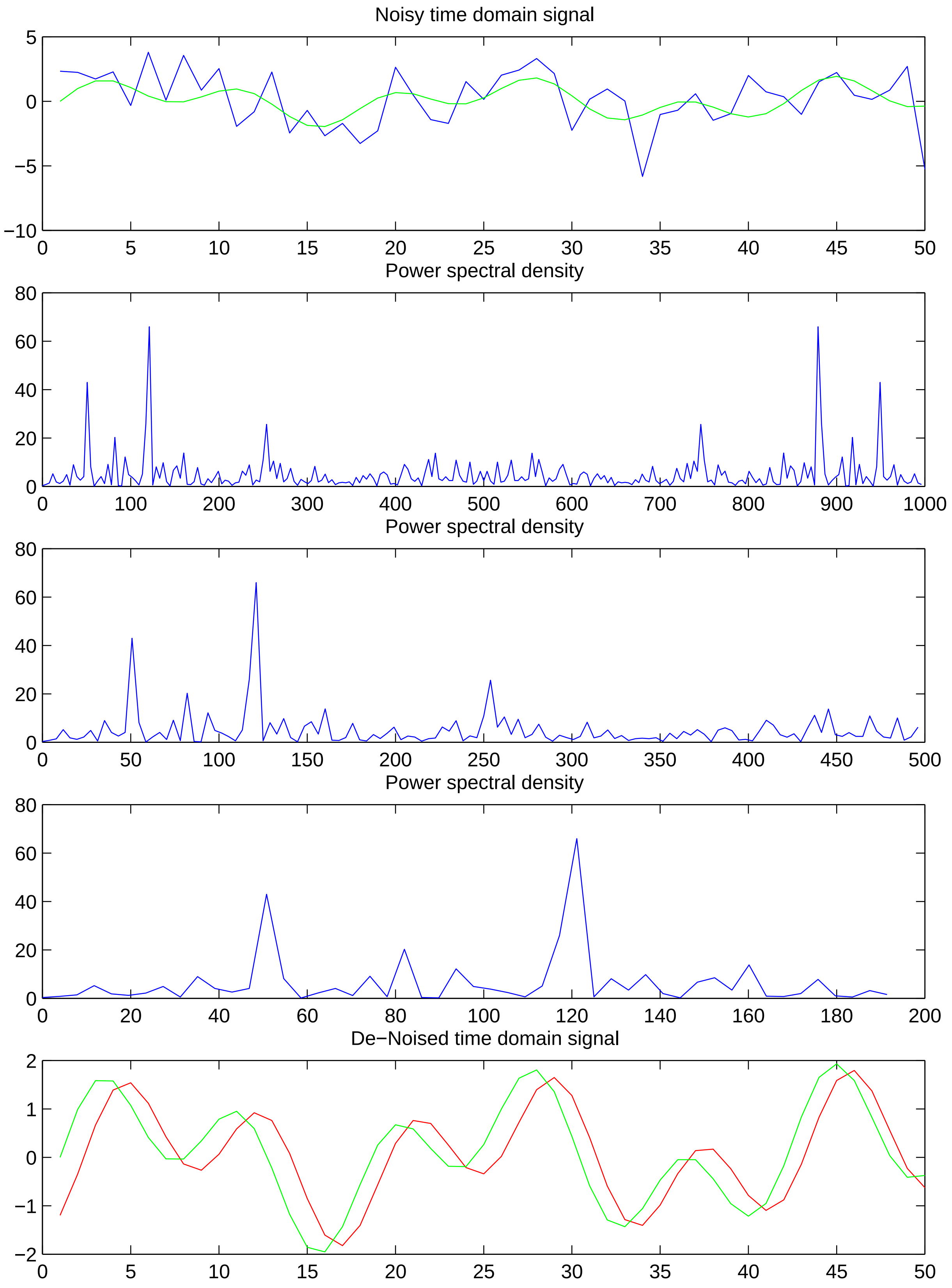


Figure 5: Sample noisy signal: 50Hz+120Hz+noise. We show the original signal, its DFT, and its inverse DFT after cleaning the DFT.