Computing the Fourier Transform

We find the expansion coefficients, $A(q)$, by computing the Fourier transform of the wave packet in x-space. The Fourier transform is defined in terms of continuous variables **x** and **q**. To implement this in **MATLAB®** we need to approximate

$$A(q) = F(\Psi(x)) = \int_{-\infty}^{+\infty} e^{-iqx}\Psi(x)dx$$

by use of the discrete equation

$$F(\Psi(x)) \approx \sum_{\{x\}} e^{-iqx}\Psi(x)\delta x$$

To do this, we create a vector with all of our x values, i.e.:

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix}$$

Similarly, we define a vector with all the values of $\Psi(x)$:

$$\vec{\Psi}(x) = \begin{pmatrix} \Psi(x_1) \\ \Psi(x_2) \\ \Psi(x_3) \\ \vdots \\ \Psi(x_n) \end{pmatrix}$$

We then define a vector for our reciprocal space:

$$\vec{q} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_n \end{pmatrix}$$

The values for $\vec{q}$ are determined by the properties of the discrete Fourier transform. Note, in general, $\vec{x}$ and $\vec{q}$ do not contain the same number of elements, i.e. $m \neq n$.

Now, the outer product of $\vec{x}$ and $\vec{q}$ is

$$\vec{x}\cdot\vec{q}^{T} = \begin{pmatrix} x_1 q_1 & x_1 q_2 & \cdots & x_1 q_m \\ x_2 q_1 & x_2 q_2 & \cdots & x_2 q_m \\ \vdots & \vdots & \ddots & \vdots \\ x_n q_1 & x_n q_2 & \cdots & x_n q_m \end{pmatrix}$$

Taking the exponential of **i** times each element of the above matrix yields:

$$\vec{\phi}_{\vec{q}}(\vec{x}) = e^{i\vec{x}\cdot\vec{q}^{T}} = \begin{pmatrix} e^{ix_1 q_1} & e^{ix_1 q_2} & \cdots & e^{ix_1 q_m} \\ e^{ix_2 q_1} & e^{ix_2 q_2} & \cdots & e^{ix_2 q_m} \\ \vdots & \vdots & \ddots & \vdots \\ e^{ix_n q_1} & e^{ix_n q_2} & \cdots & e^{ix_n q_m} \end{pmatrix}$$

Note, the **MATLAB®** command `` exp" takes the exponent of each element in a matrix. Then,

$$
\begin{aligned}
\vec{A}(q) &= \sum_{\{x\}} \vec{\phi}_{\bar{q}}(x)\vec{\Psi}(x)\delta x \\
&= \vec{\phi}_{\bar{q}}^{\dagger}(x)\cdot\vec{\Psi}(x)\delta x \\
&= \begin{pmatrix} e^{ix_1q_1} & e^{ix_1q_2} & \cdots & e^{ix_1q_m} \\ e^{ix_2q_1} & e^{ix_2q_2} & \cdots & e^{ix_2q_m} \\ \vdots & \vdots & \ddots & \vdots \\ e^{ix_nq_1} & e^{ix_nq_2} & \cdots & e^{ix_nq_m} \end{pmatrix}^{\dagger} \cdot \begin{pmatrix} \Psi(x_1) \\ \Psi(x_2) \\ \Psi(x_3) \\ \vdots \\ \Psi(x_n) \end{pmatrix} \delta x \\
&= \begin{pmatrix} e^{-iq_1x_1} & e^{-iq_1x_2} & \cdots & e^{-iq_1x_n} \\ e^{-iq_2x_1} & e^{-iq_2x_2} & \cdots & e^{-iq_2x_n} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-iq_mx_1} & e^{-iq_mx_2} & \cdots & e^{-iq_mx_n} \end{pmatrix} \cdot \begin{pmatrix} \Psi(x_1) \\ \Psi(x_2) \\ \Psi(x_3) \\ \vdots \\ \Psi(x_n) \end{pmatrix} \delta x \\
&= \begin{pmatrix} \Psi(x_1)e^{-iq_1x_1} + \Psi(x_2)e^{-iq_1x_2} + \cdots + \Psi(x_n)e^{-iq_1x_n} \\ \Psi(x_1)e^{-iq_2x_1} + \Psi(x_2)e^{-iq_2x_2} + \cdots + \Psi(x_n)e^{-iq_2x_n} \\ \vdots \\ \Psi(x_1)e^{-iq_mx_1} + \Psi(x_2)e^{-iq_mx_2} + \cdots + \Psi(x_n)e^{-iq_mx_n} \end{pmatrix} \delta x
\end{aligned}
$$

i.e. a vector indexed by **q** which contains the sum we defined to be an approximation to the Fourier transform, known as the Discrete Fourier Transform (DFT).