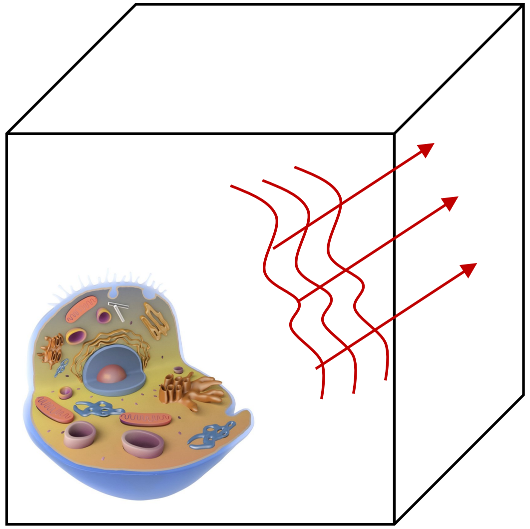
Machine Learning in Imaging

BME 590L
Roarke Horstmeyer

Lecture 3b: Discrete Mathematics, Part II

ML+Imaging pipeline introduction

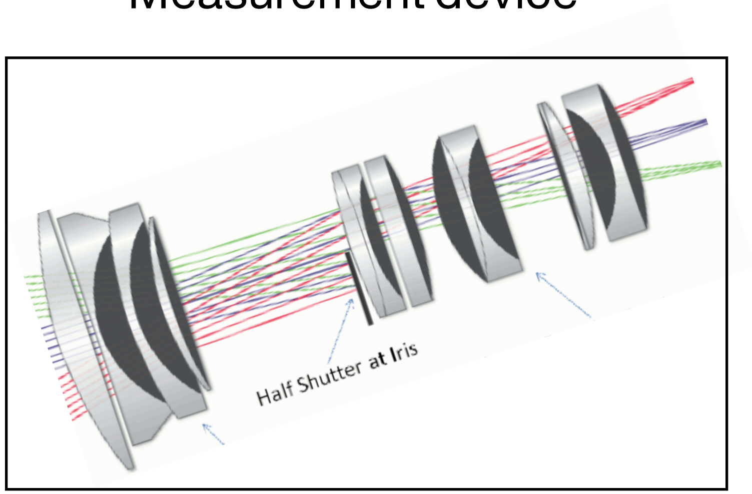
Real World



Continuous complex fields

(last week)

Measurement device

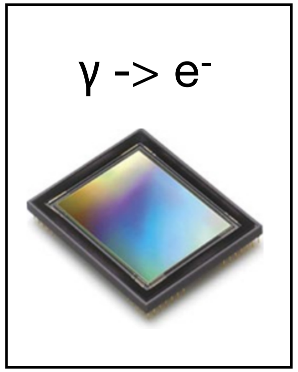


Black box transformations

- Convolution
- Fourier Transform

(last week)

Digitization

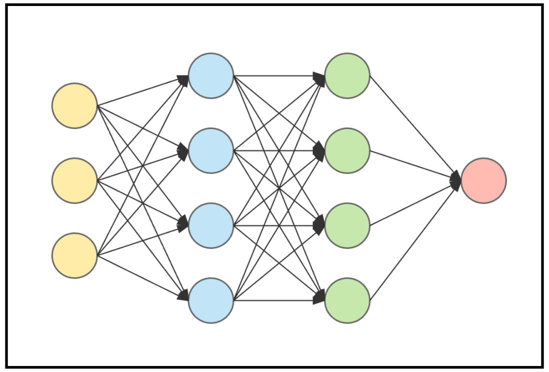


Sampling Theorem

Discrete math & Linear algebra

(this class)

Machine Learning



Optimization (this class)

Linear classification

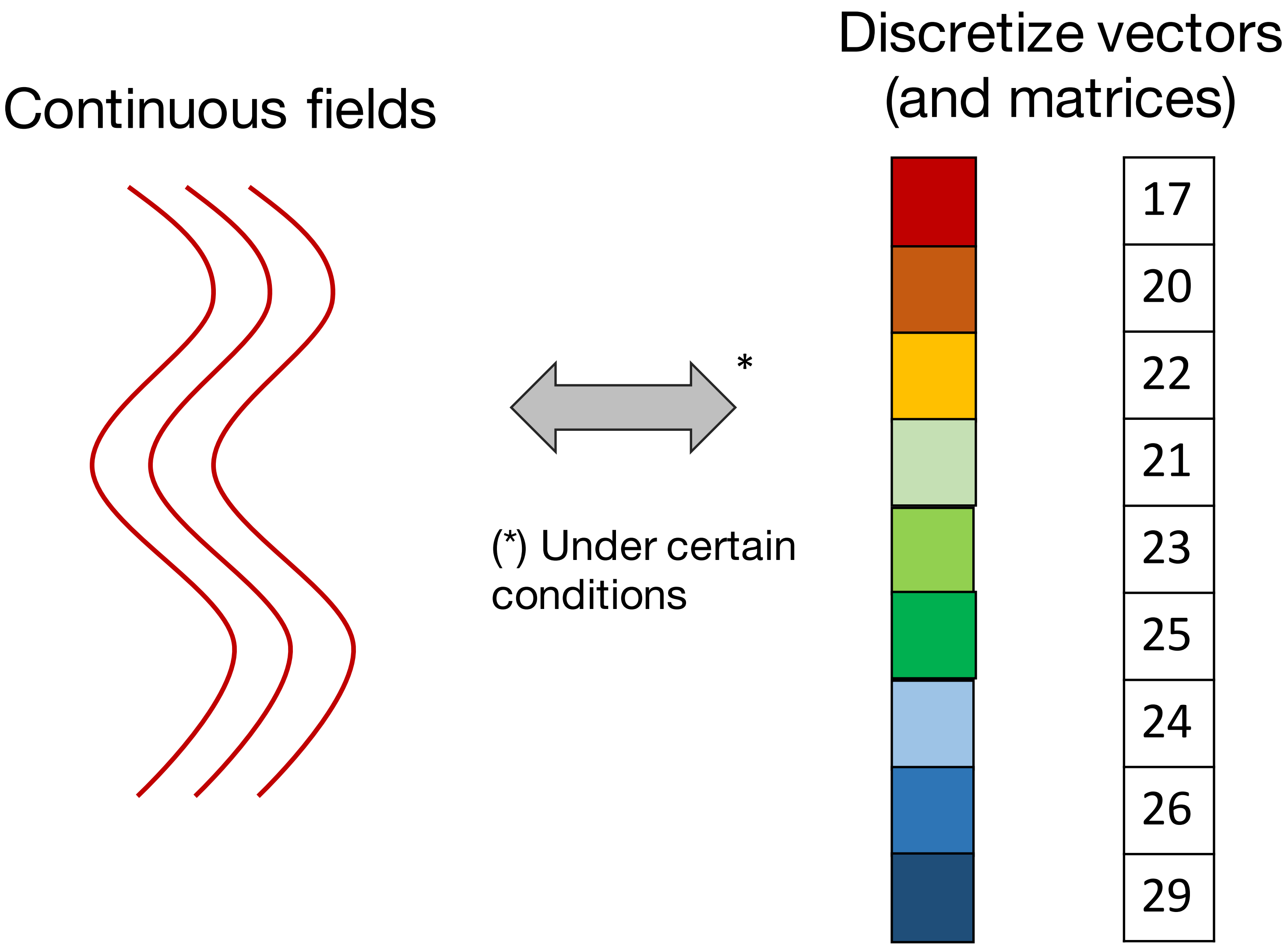
Logistic classifier

Neural networks

Convolutional NN's

(next few weeks)

What does the Sampling Theorem mean for us?



Discrete convolution

$$V(x_o) = \int_{-\infty}^{\infty} U(x_i)h(x_o - x_i)dx_i$$

$$\downarrow$$

$$v[x_0] = \sum_{x_i=-M}^{M} u[x_i]h[x_o - x_i]$$

Example 2: Find the convolution of the two sequences x[n] and h[n] given by,

$$x[k] = [\underset{\uparrow}{3}\ 1\ 2] \qquad h[k] = [3\ \underset{\uparrow}{2}\ 1]$$

k:	-2	-1	0	1	2	3	4	5
x[k]:			3	$\boxed{1}$	$\boxed{2}$	$\boxed{\ }$		
h[-k]:	1	2	3					
h[1-k]:		1	2	3				
h[2-k]:			1	2	3			
h[3-k]:				$\boxed{1}$	$\boxed{2}$	$\boxed{3}$		
h[4-k]:					1	2	3	
h[5-k]:						1	2	3
y:			9	6+3	3+2+6	1+4+0		

y: [9 9 11 5 2 0]

Discrete convolution

$$V(x_o) = \int_{-\infty}^{\infty} U(x_i) h(x_o - x_i) dx_i$$

$$\downarrow$$

$$v[x_0] = \sum_{x_i=-M}^{M} u[x_i] h[x_o - x_i]$$

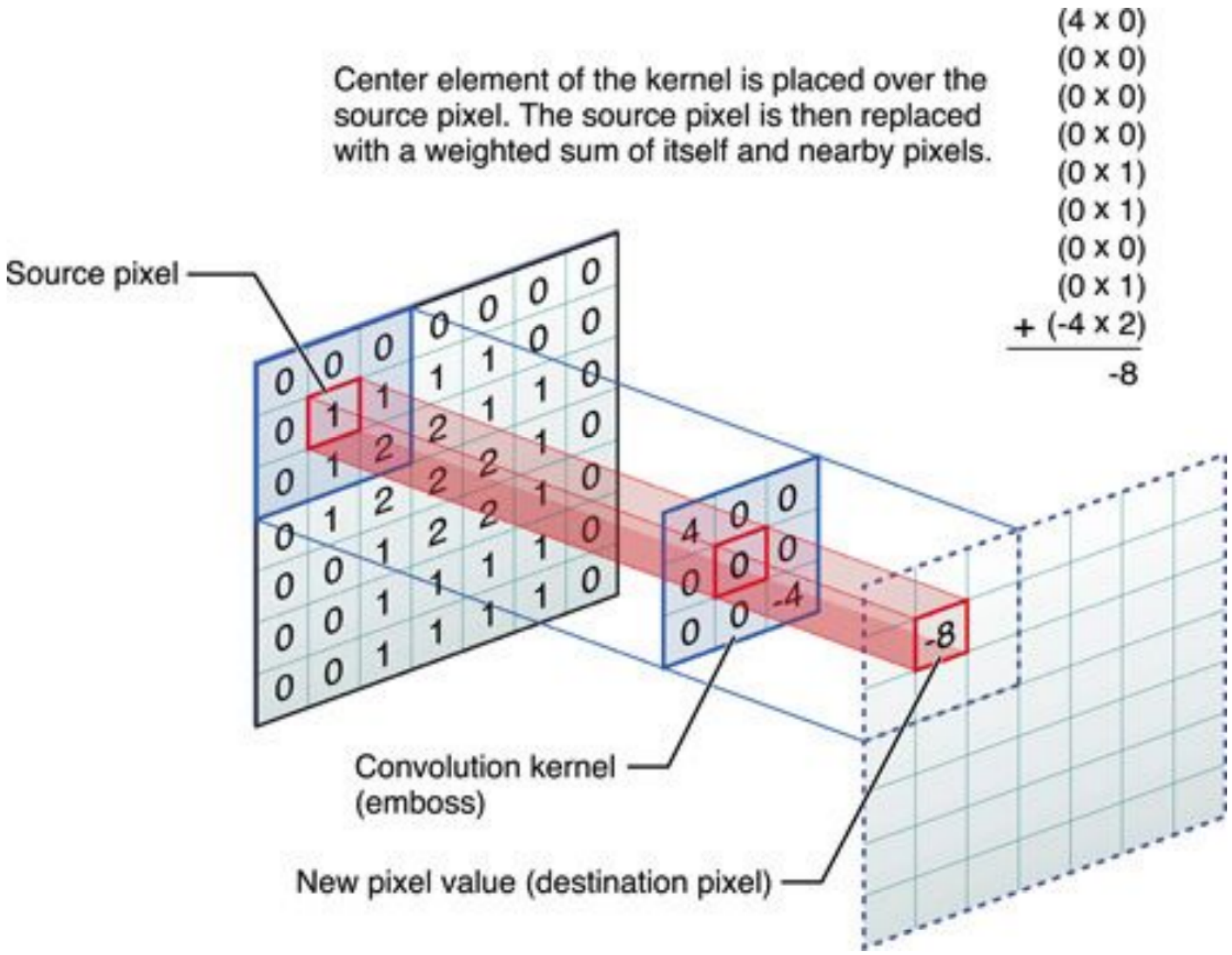
Discrete 2D convolution

$$V(x_o, y_o) = \iint_{-\infty}^{\infty} U(x_i, y_i) h(x_o - x_i, y_o - y_i) dx_i dy_i$$

$$\downarrow$$

$$v[x_0, y_o] = \sum_{y_i=-L}^{L} \sum_{x_i=-M}^{M} u[x_i, y_i] h[x_o - x_i, y_o - y_i]$$

Discrete 2D convolution

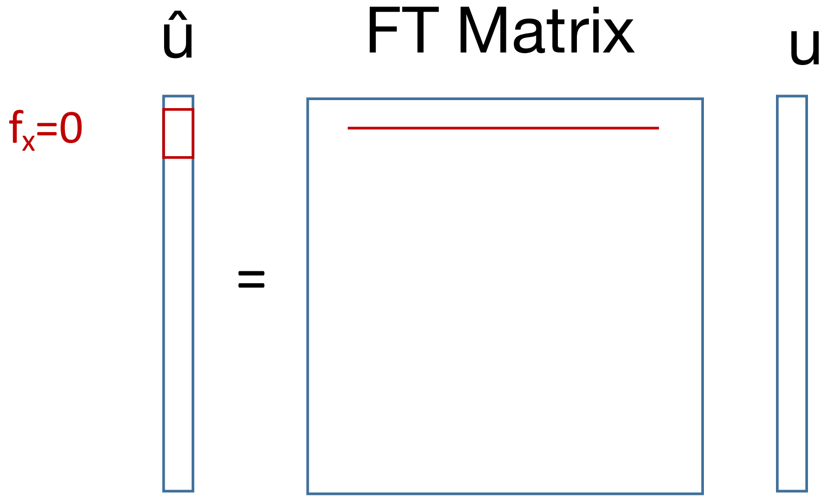


https://www.psi.toronto.edu/~jimmy/ece521/Tut1.pdf

Discrete Fourier Transforms

$$\hat{U}(f_x) = \int_{-\infty}^{\infty} U(x) \exp\left(-2\pi i (f_x x)\right) dx$$

$$\hat{u}[f_x] = \sum_{x=0}^{M-1} u[x] \exp\left(-2\pi i f_x x / M\right)$$



Inner product of u with different complex expon.

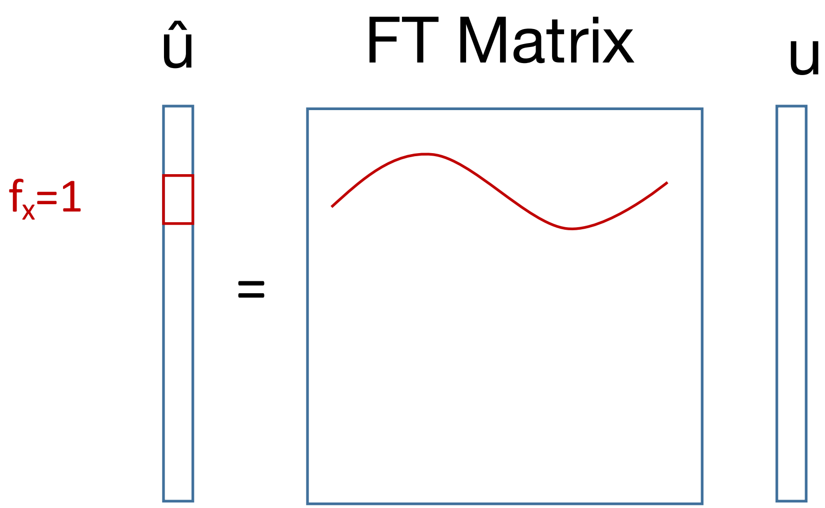
- np.fft(u), np.fftshift(np.fft(np.ifftshift(u)))
- fft = fast Fourier transform, much more comp. efficient than matrix multiplication!

Discrete Fourier Transforms

$$\hat{U}(f_x) = \int_{-\infty}^{\infty} U(x) \exp\left(-2\pi i (f_x x)\right) dx$$

$$\hat{u}[f_x] = \sum_{x=0}^{M-1} u[x] \exp\left(-2\pi i f_x x / M\right)$$

Inner product of u with different complex expon.

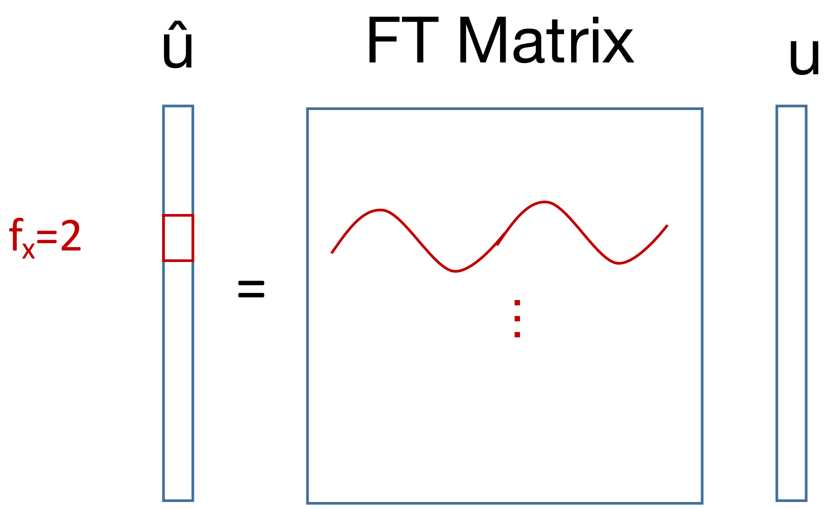


- np.fft(u), np.fftshift(np.fft(np.ifftshift(u)))
- fft = fast Fourier transform, much more comp. efficient than matrix multiplication!

Discrete Fourier Transforms

$$\hat{U}(f_x) = \int_{-\infty}^{\infty} U(x) \exp\left(-2\pi i (f_x x)\right) dx$$

$$\hat{u}[f_x] = \sum_{x=0}^{M-1} u[x] \exp\left(-2\pi i f_x x / M\right)$$



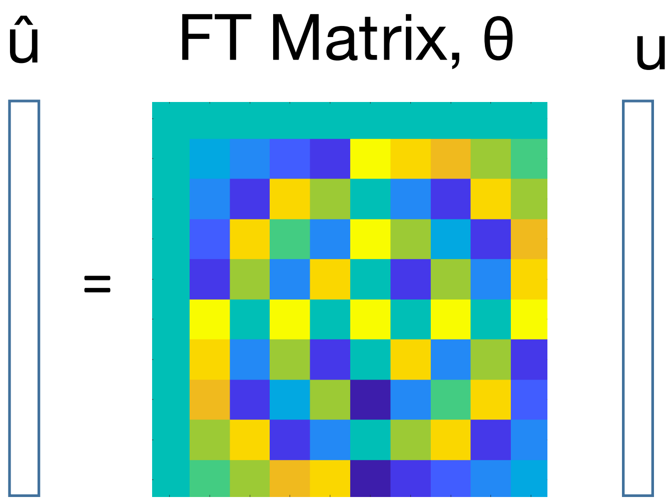
Inner product of u with different complex expon.

- np.fft(u), np.fftshift(np.fft(np.ifftshift(u)))
- fft = fast Fourier transform, much more comp. efficient than matrix multiplication!

Discrete Fourier Transforms

$$\hat{U}(f_x) = \int_{-\infty}^{\infty} U(x) \exp\left(-2\pi i (f_x x)\right) dx$$

$$\hat{u}[f_x] = \sum_{x=0}^{M-1} u[x] \exp\left(-2\pi i f_x x / M\right)$$



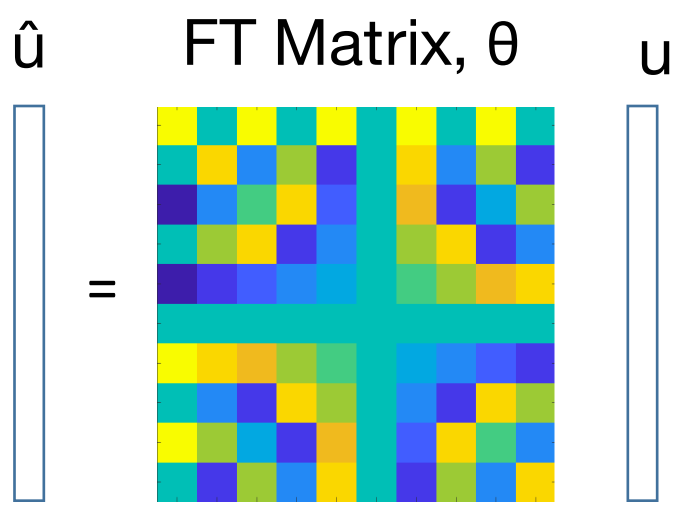
np.fft(np.eye(10))

Treats 1st entry of û as f_x=0

Discrete Fourier Transforms

$$\hat{U}(f_x) = \int_{-\infty}^{\infty} U(x) \exp\left(-2\pi i (f_x x)\right) dx$$

$$\hat{u}[f_x] = \sum_{x=0}^{M-1} u[x] \exp\left(-2\pi i f_x x / M\right)$$



np.fftshift(np.fft(np.ifftshift(np.eye(10))))

Treats middle entry of û as f_x=0

Discrete convolution theorem

Convolution theorem. If $\mathcal{F}\{g(x, y)) = G(f_X, f_Y)$ and $\mathcal{F}\{h(x, y)\} = H(f_X, f_Y)$, then

$$\mathcal{F}\left\{\iint_{-\infty}^{\infty} g(\xi, \eta)\, h(x - \xi, y - \eta)\, d\xi\, d\eta\right\} = G(f_X, f_Y)\, H(f_X, f_Y). \qquad \text{(2-15)}$$

Discrete convolution theorem

Convolution theorem. If $\mathcal{F}\{g(x, y)) = G(f_X, f_Y)$ and $\mathcal{F}\{h(x, y)\} = H(f_X, f_Y)$, then

Continuous

$$\mathcal{F}\left\{\iint_{-\infty}^{\infty} g(\xi, \eta)\, h(x-\xi, y-\eta)\, d\xi\, d\eta\right\} = G(f_X, f_Y)\, H(f_X, f_Y). \qquad (2\text{-}15)$$

If $\mathcal{F}[g[x,y]] = G[f_x, f_y]$ and $\mathcal{F}[h[x,y]] = H[f_x, f_y]$, and if we know that

Discrete

$$g[x,y] * h[x,y] = \sum_{l=-L}^{L} \sum_{m=-M}^{M} g[m,l]h[x-m, y-l],$$

then from the Convolution Theorem we have,

$$\mathcal{F}[g[x,y] * h[x,y]] = G[f_x, f_y]H[f_x, f_y]$$

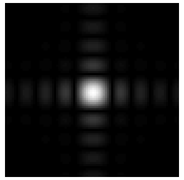
Discrete convolution theorem example

Input image

$U_1(x,y)$

$*$

Convolution filter h

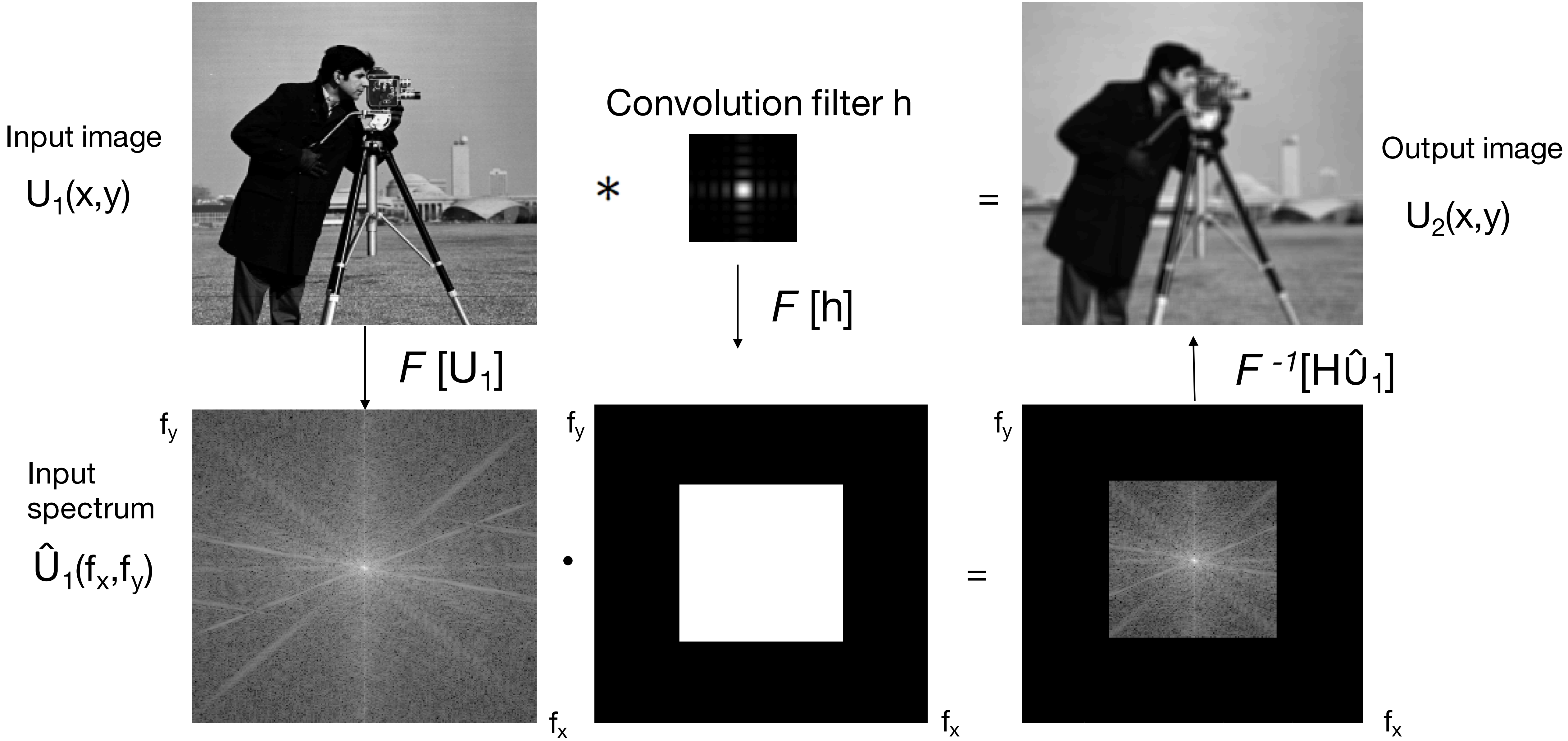


$=$

Output image

$U_2(x,y)$

Discrete convolution theorem example



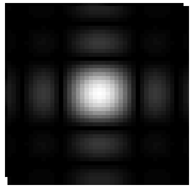
Discrete convolution theorem example

Input image

$U_1(x,y)$

Convolution filter h

$*$

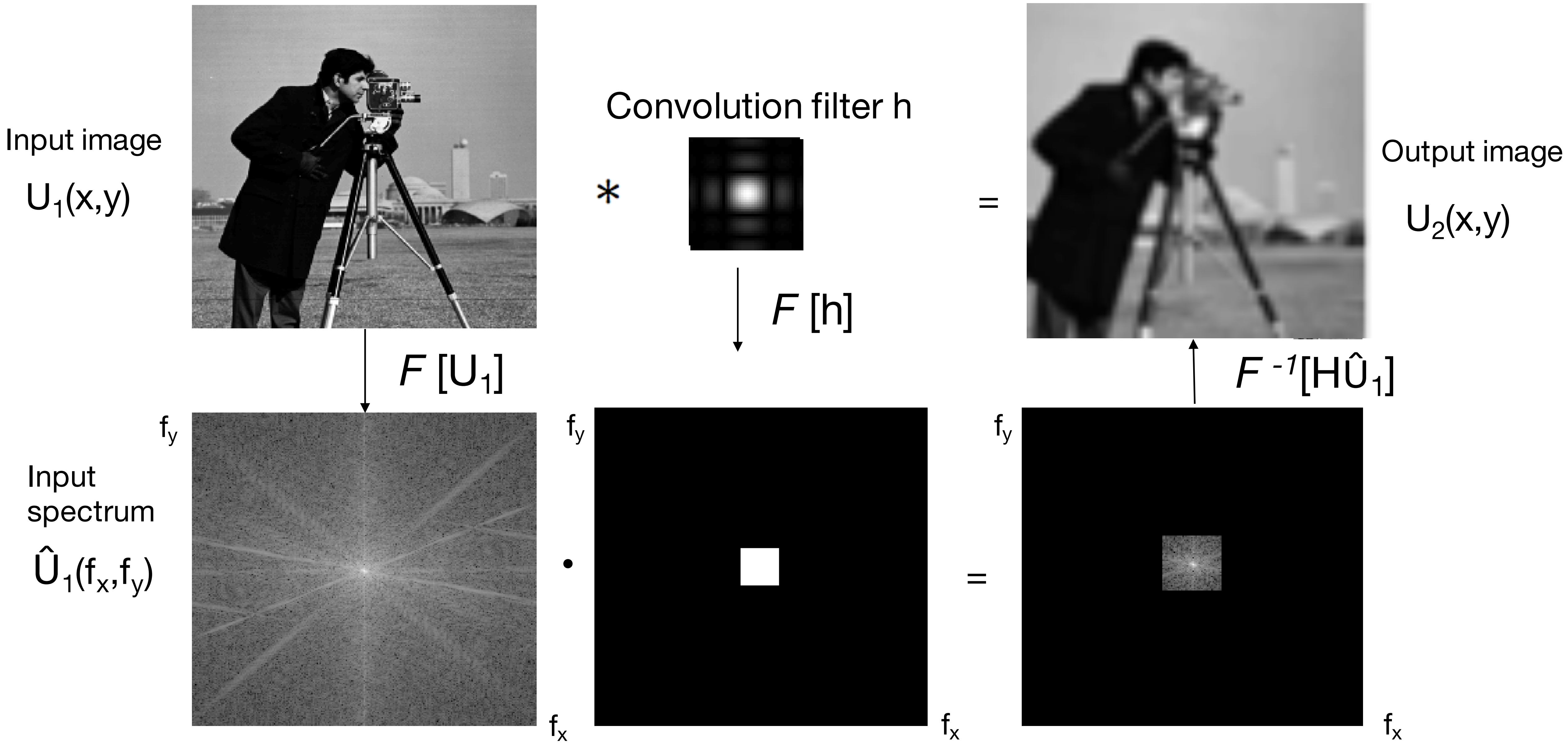


$=$

Output image

$U_2(x,y)$

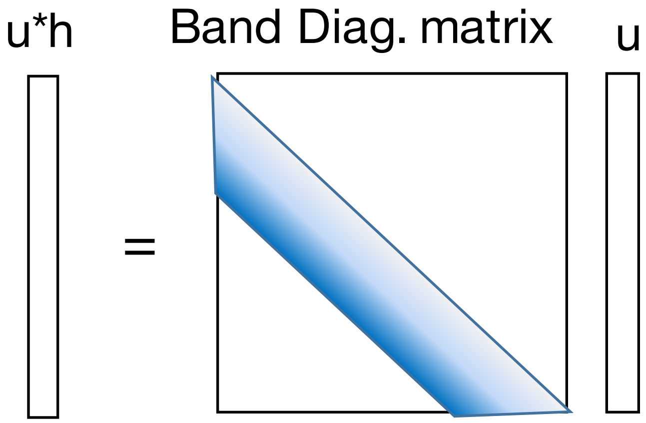
Discrete convolution theorem example



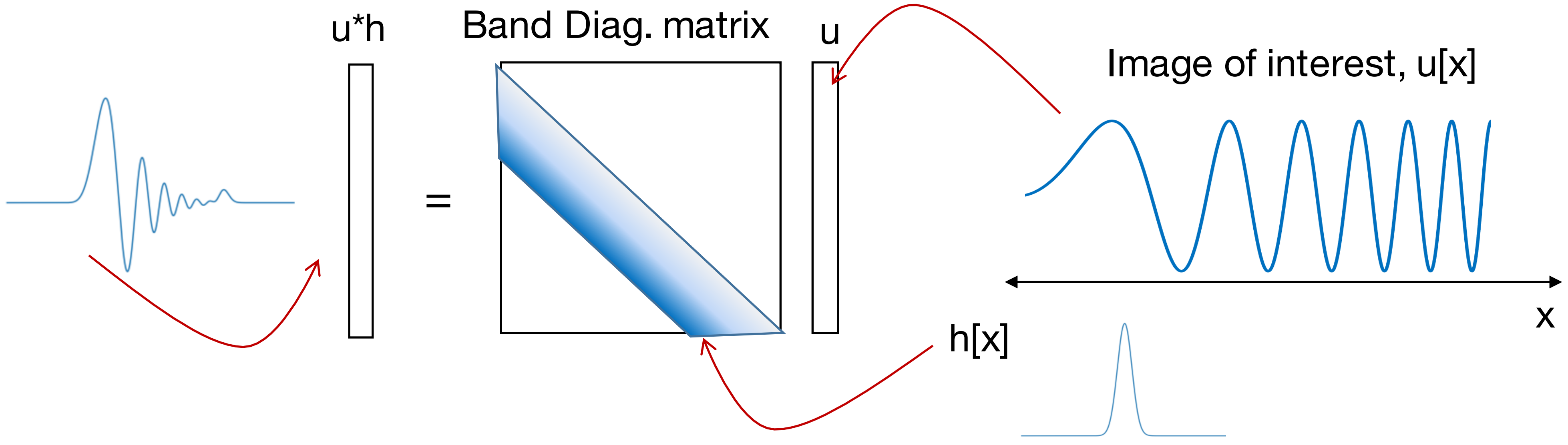
Convolutions as a big matrix multiplication

$$(u * h)[x] = \sum_{m=0}^{N+M-2} u[m]h[x-m] \longrightarrow y = u * h = \begin{bmatrix} h_1 & 0 & \dots & 0 & 0 \\ h_2 & h_1 & \dots & \vdots & \vdots \\ h_3 & h_2 & \dots & 0 & 0 \\ \vdots & h_3 & \dots & h_1 & 0 \\ h_{m-1} & \vdots & \dots & h_2 & h_1 \\ h_m & h_{m-1} & \vdots & \vdots & h_2 \\ 0 & h_m & \dots & h_{m-2} & \vdots \\ 0 & 0 & \dots & h_{m-1} & h_{m-2} \\ \vdots & \vdots & \vdots & h_m & h_{m-1} \\ 0 & 0 & 0 & \dots & h_m \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_n \end{bmatrix}$$

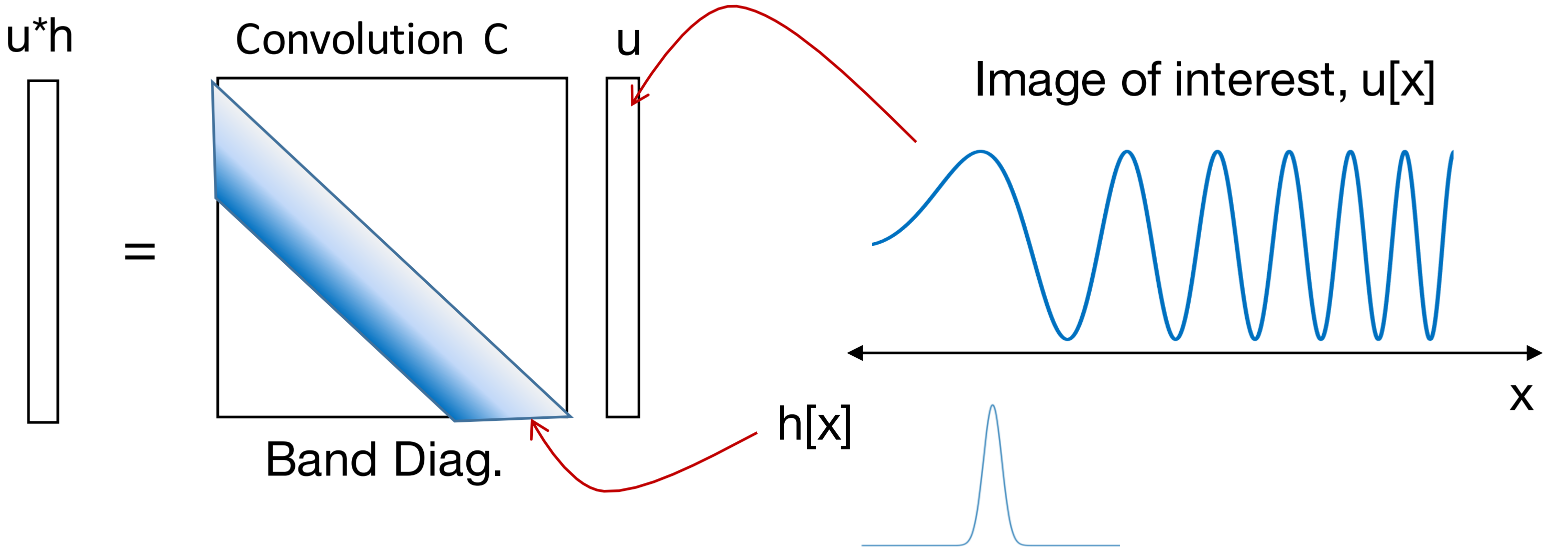
Convolutions as a big matrix multiplication



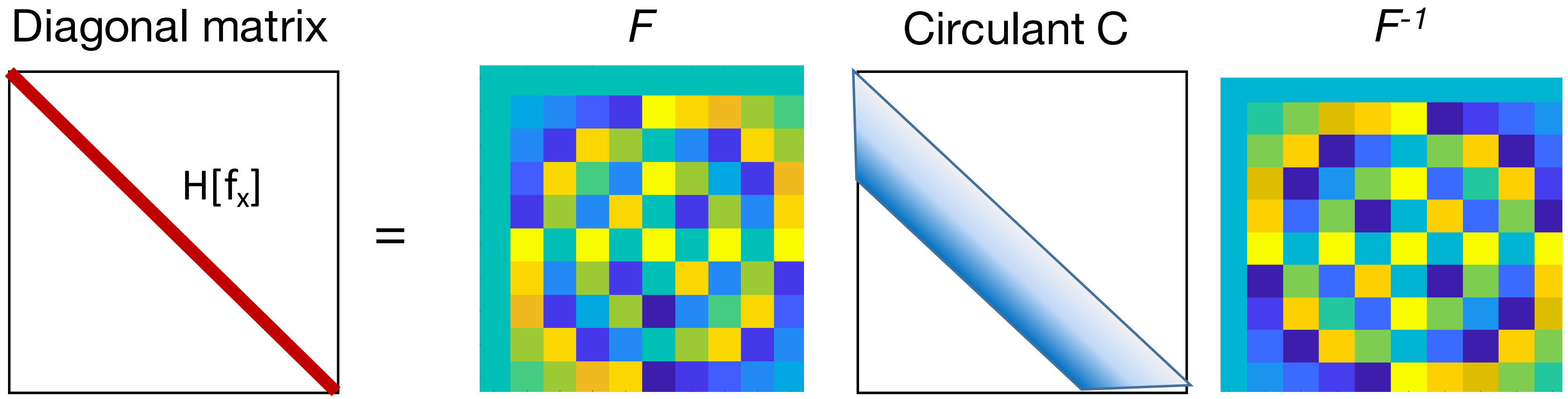
Convolutions as a big matrix multiplication



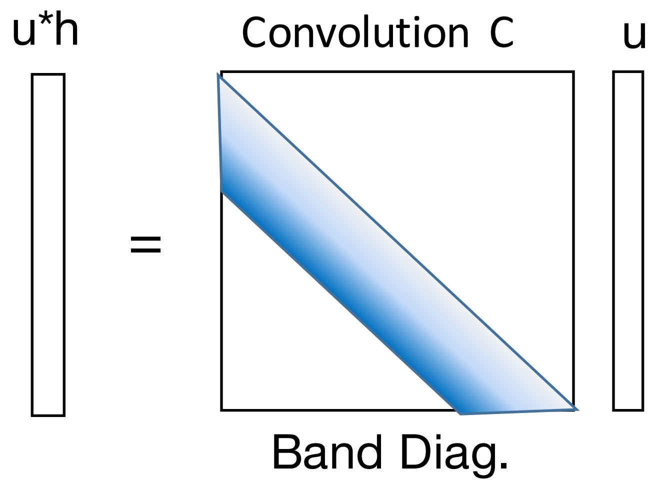
Convolutions as a big matrix multiplication

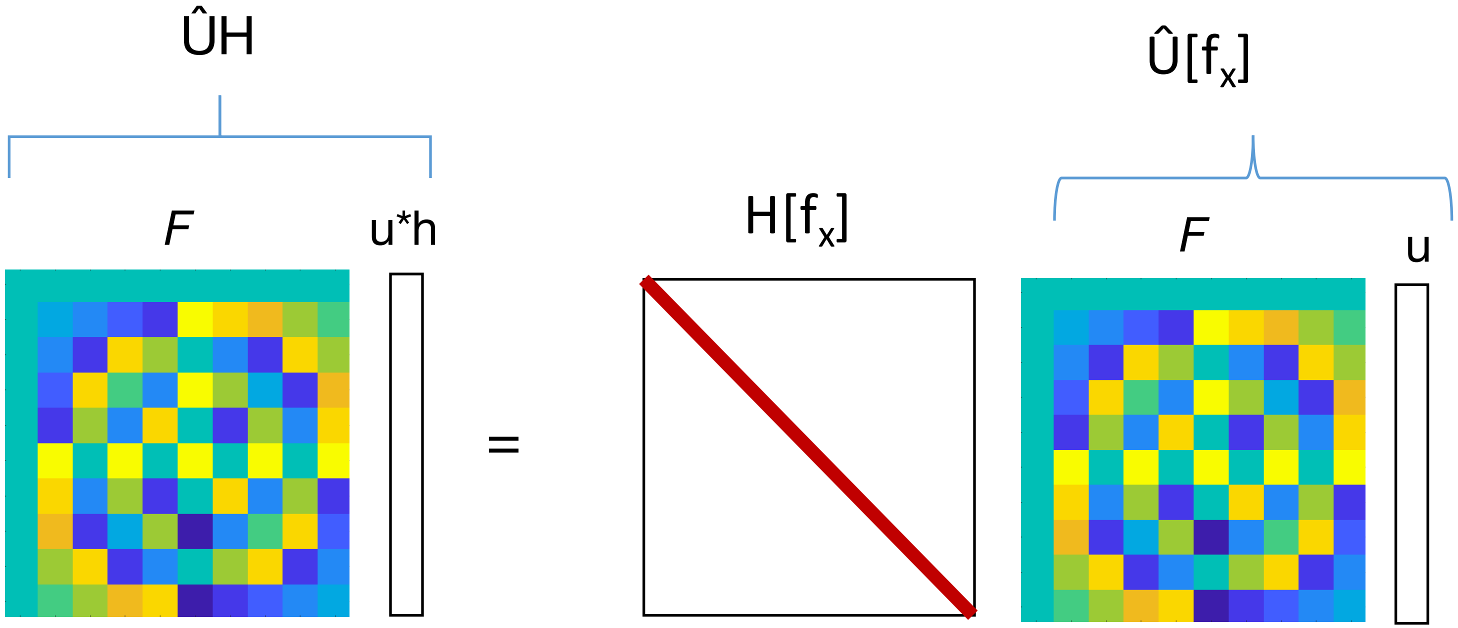


Discrete Fourier transform diagonalizes convolution matrix with $H[f_x]$ along diag.



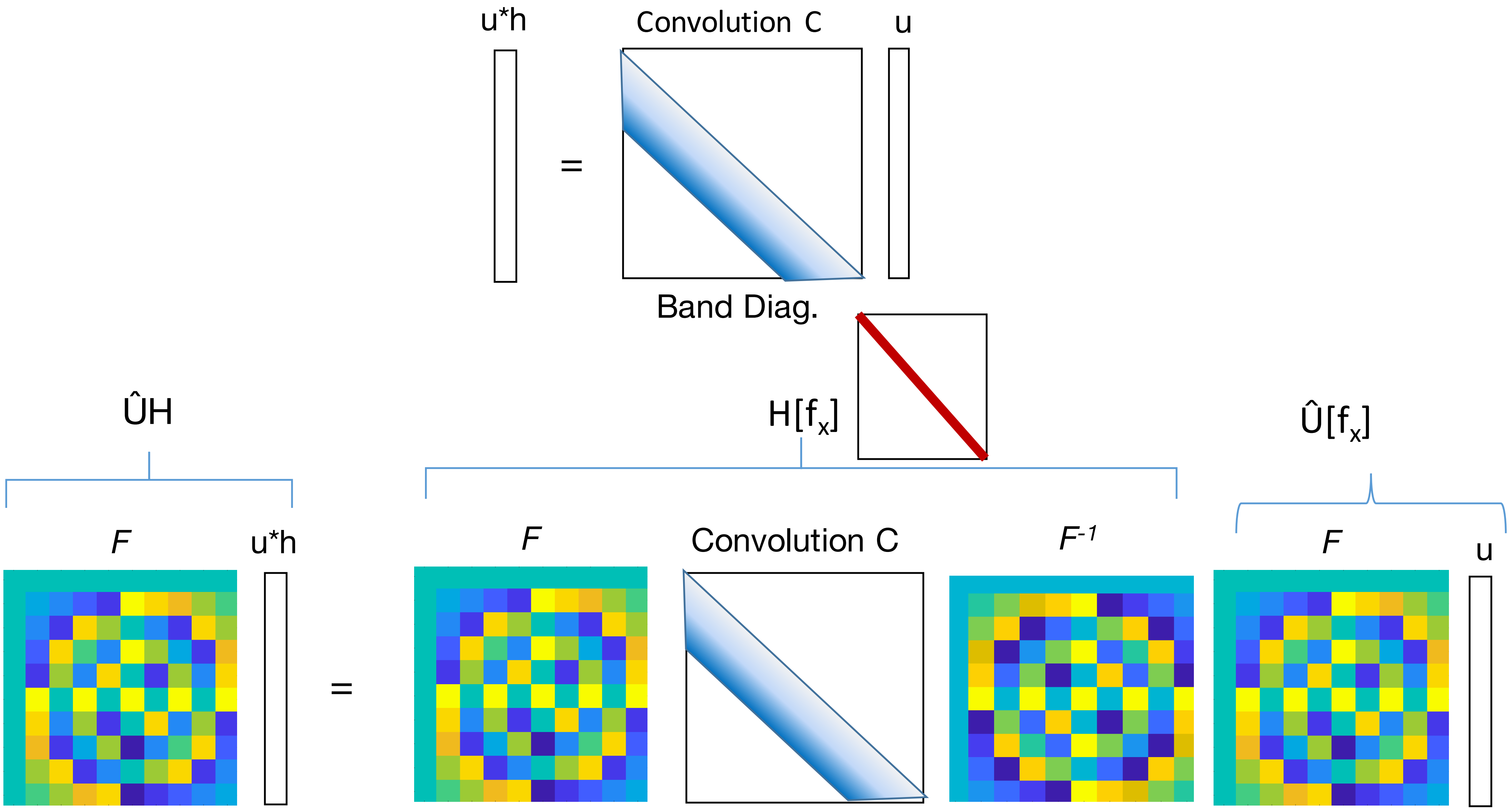
Convolutions as a big matrix multiplication



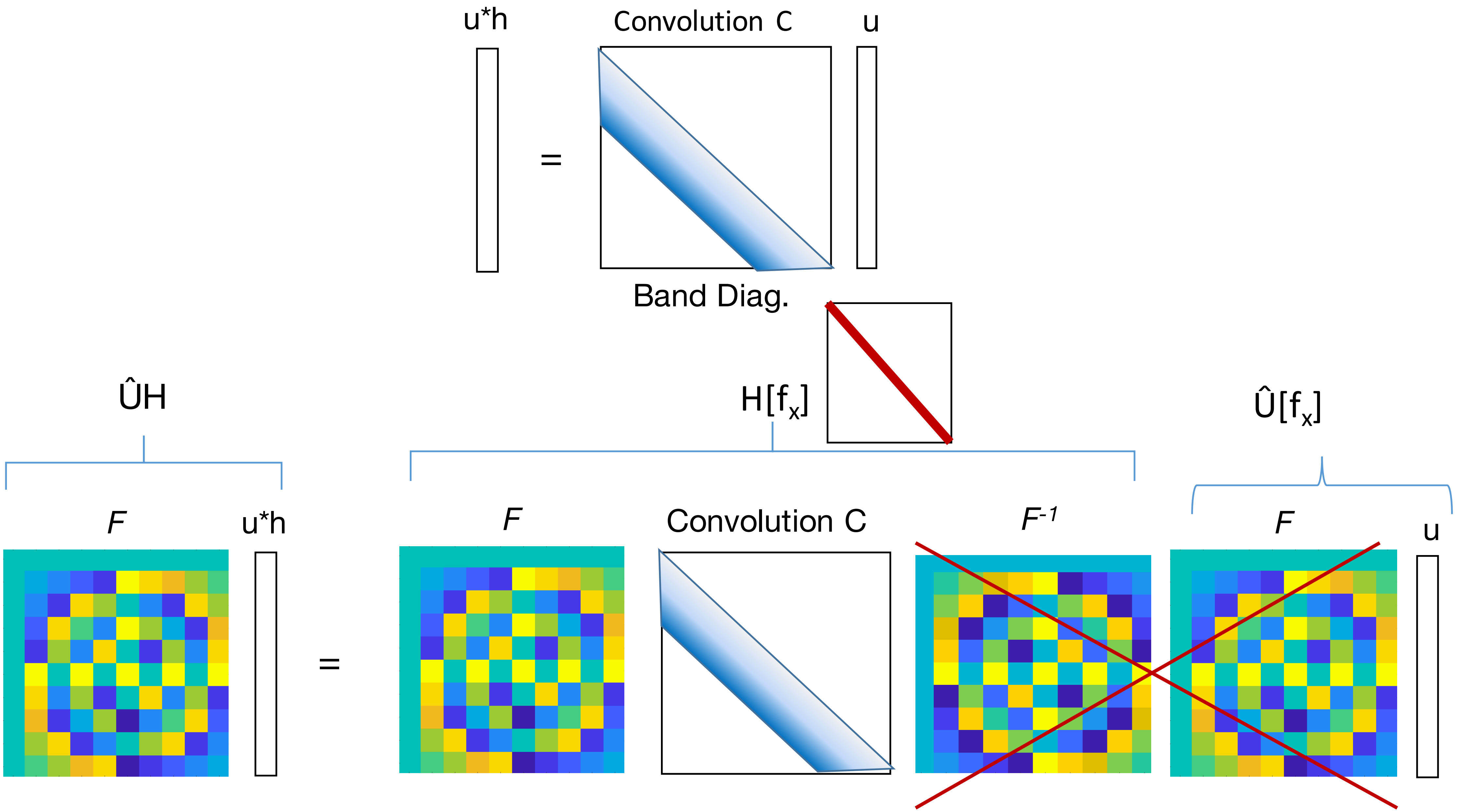


- Convolution achieved by multiplication in Fourier domain

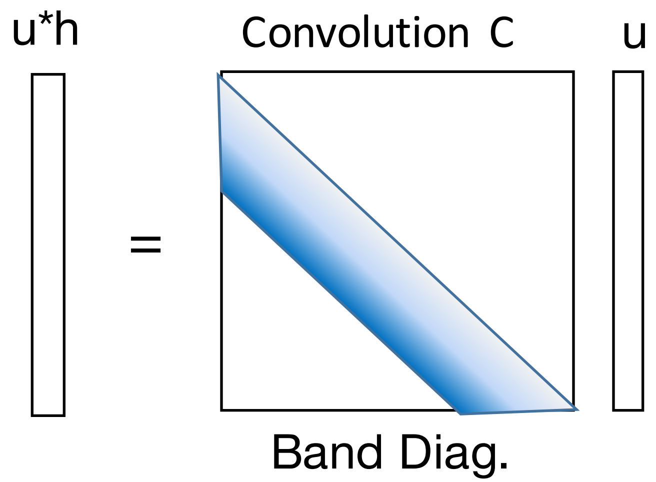
Convolutions as a big matrix multiplication

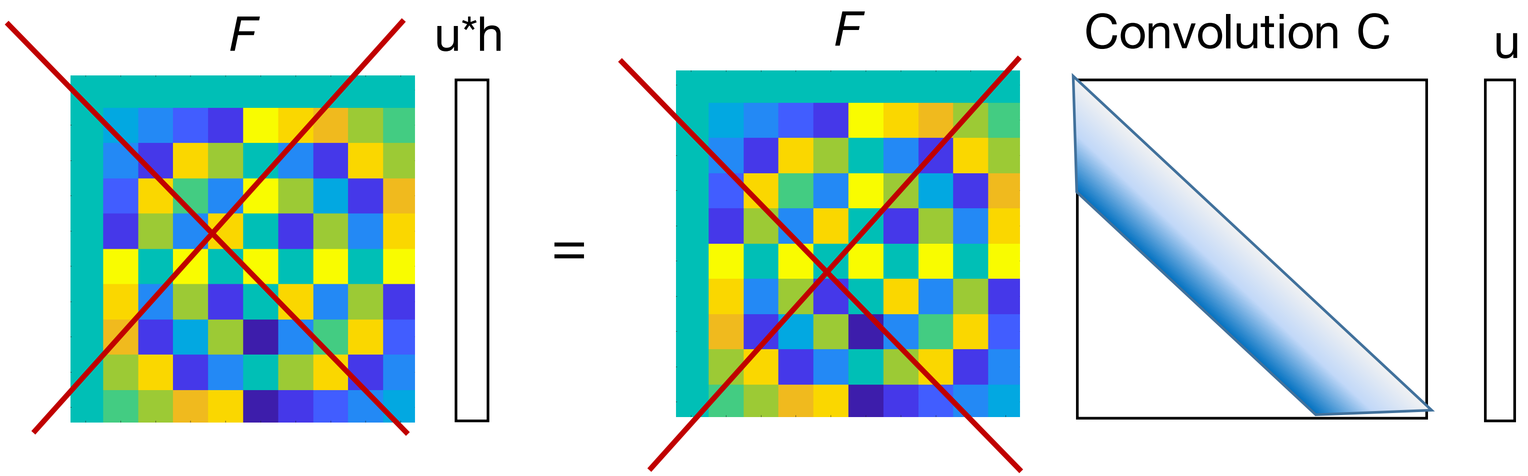


Convolutions as a big matrix multiplication

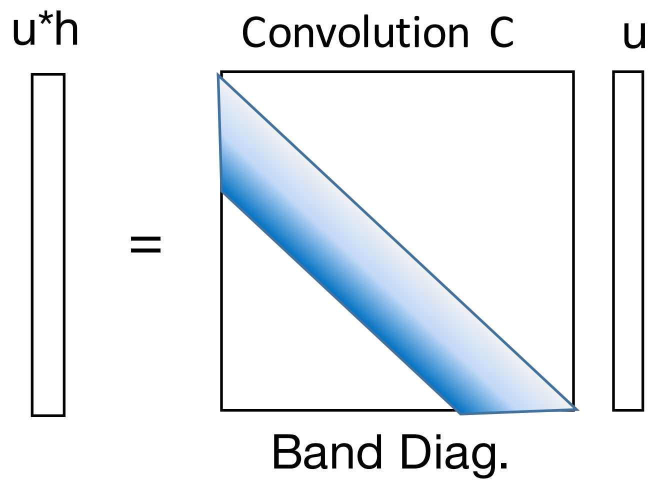


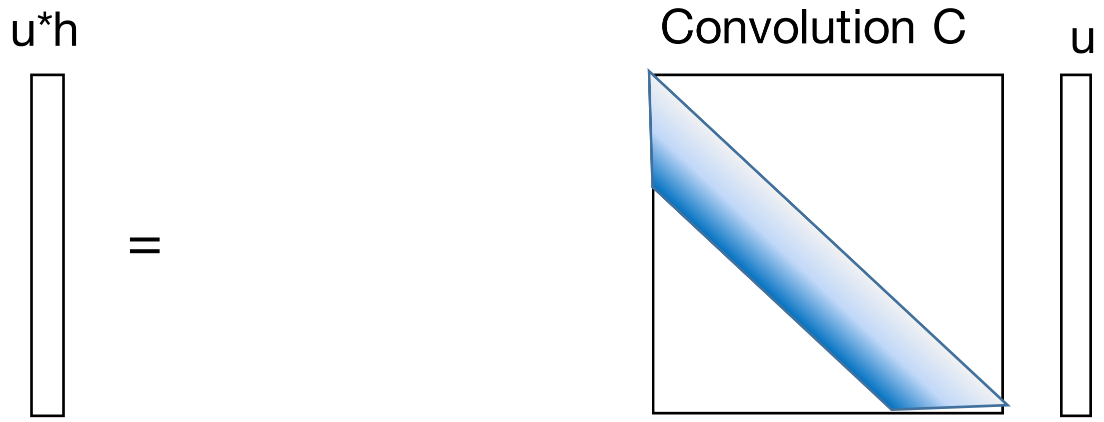
Convolutions as a big matrix multiplication





Convolutions as a big matrix multiplication





Last thing – matrix and vector derivatives

$$\mathbf{u} = \mathbf{W}\mathbf{v}$$

$$\frac{d\mathbf{u}}{d\mathbf{v}} =$$

Last thing – matrix and vector derivatives

$$\mathbf{u} = \mathbf{W}\mathbf{v}$$

$$\frac{d\mathbf{u}}{d\mathbf{v}} =$$

$$\mathbf{u}_3 = W_{3,1}v_1 + W_{3,2}v_2 + ... + W_{3,M}v_M$$

$$\frac{\partial u_3}{\partial v_2} = \frac{\partial}{\partial v_2}(W_{3,1}v_1 + W_{3,2}v_2 + ... + W_{3,M}v_M) = \frac{\partial}{\partial v_2}W_{3,2}v_2 = W_{3,2}$$

Last thing – matrix and vector derivatives

$$\mathbf{u} = \mathbf{W}\mathbf{v}$$

$$\frac{d\mathbf{u}}{d\mathbf{v}} =$$

$$\mathbf{u}_3 = W_{3,1}v_1 + W_{3,2}v_2 + ... + W_{3,M}v_M$$

$$\frac{\partial u_3}{\partial v_2} = \frac{\partial}{\partial v_2}(W_{3,1}v_1 + W_{3,2}v_2 + ... + W_{3,M}v_M) = \frac{\partial}{\partial v_2}W_{3,2}v_2 = W_{3,2}$$

$$\frac{\partial u_i}{\partial v_j} = W_{i,j}$$

$$\frac{d\mathbf{u}}{d\mathbf{v}} = \mathbf{W}$$

Last thing – matrix and vector derivatives

$$\mathbf{u} = \mathbf{W}\mathbf{v}$$

$$\frac{d\mathbf{u}}{d\mathbf{v}} =$$

$$\mathbf{u}_3 = W_{3,1}v_1 + W_{3,2}v_2 + ... + W_{3,M}v_M$$

$$\frac{\partial u_3}{\partial v_2} = \frac{\partial}{\partial v_2}(W_{3,1}v_1 + W_{3,2}v_2 + ... + W_{3,M}v_M) = \frac{\partial}{\partial v_2}W_{3,2}v_2 = W_{3,2}$$

$$\frac{\partial u_i}{\partial v_j} = W_{i,j}$$

$$\frac{d\mathbf{u}}{d\mathbf{v}} = \mathbf{W}$$

- When confused, write out one entry, solve derivative and generalize
- Use dimensionality to help (if **x** has N elements, and **y** has M, then d**y/**d**x** must be NxM
- Take advantage of *The Matrix Cookbook*:
  - https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf