

Computational Methods (PHYS 2030)

Instructors: Prof. Christopher Bergevin (cberge@yorku.ca)

Schedule: Lecture: MWF 11:30-12:30 (CLH M)

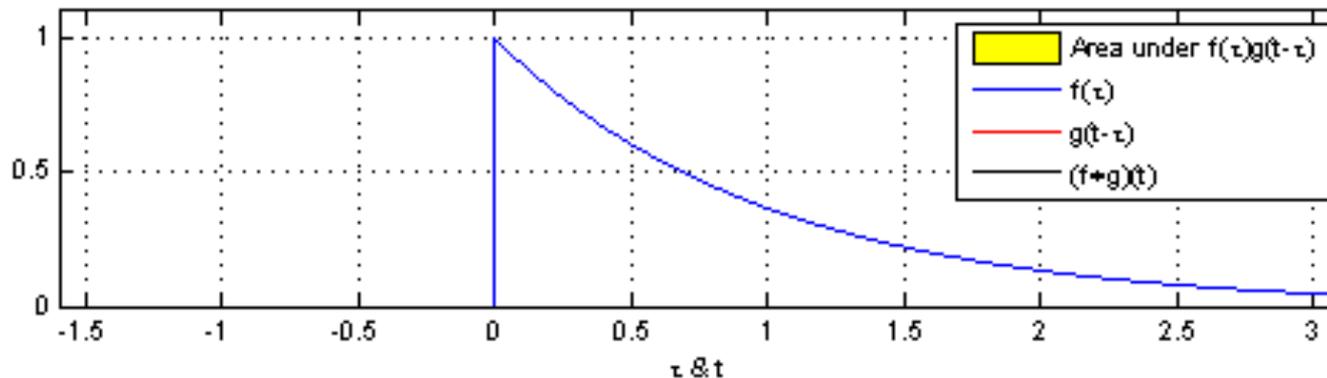
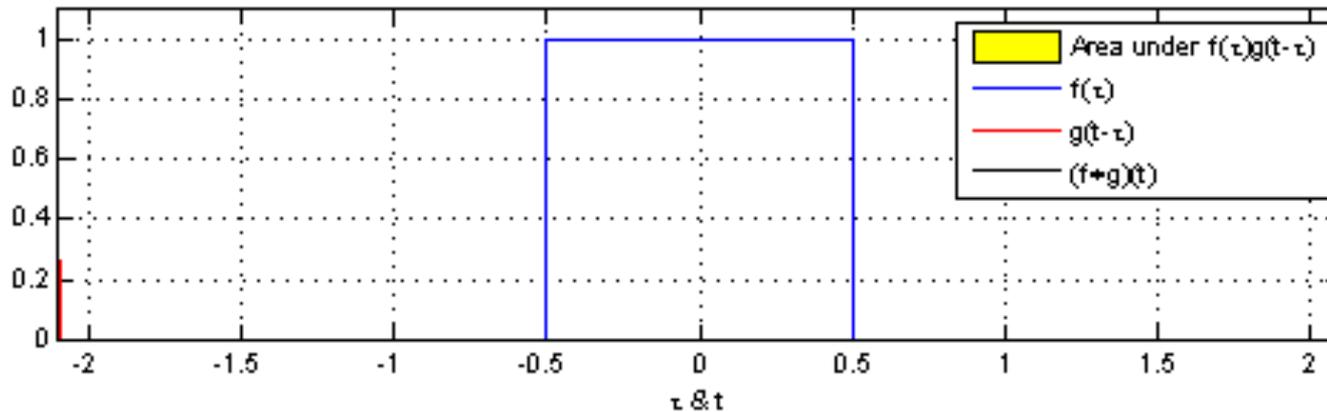
Website: <http://www.yorku.ca/cberge/2030W2018.html>

Convolution

- A deceptively complex quantity. Consider two functions $p(t)$ and $q(t)$. Then the convolution is defined as:

$$p \otimes q = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} p(\tau)q(t - \tau) d\tau$$

Note that this quantity will be a function of t (τ is the 'sliding' variable here)

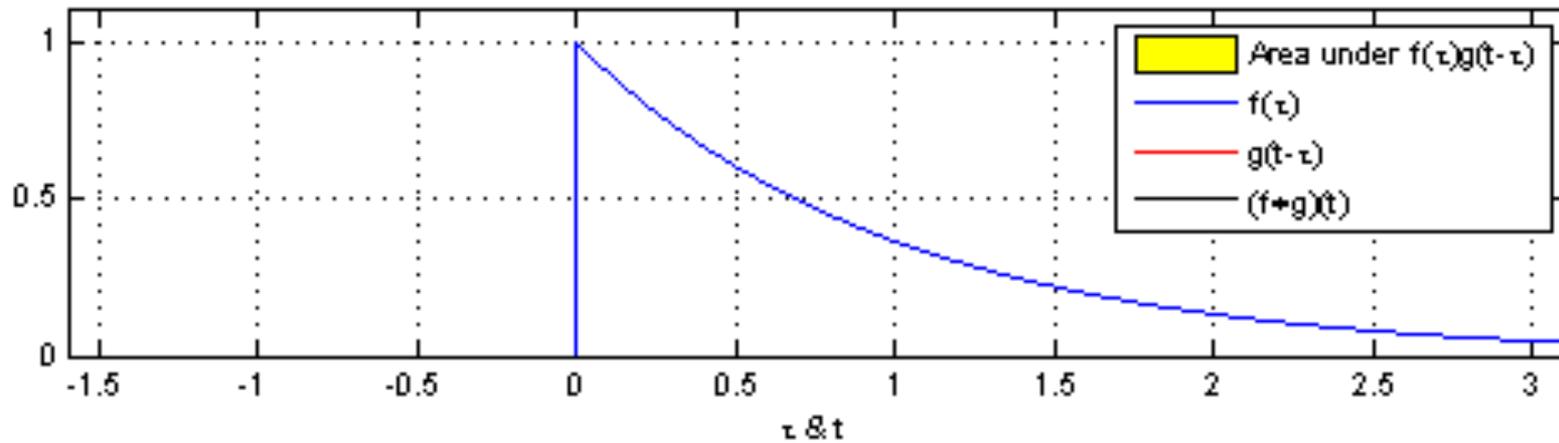
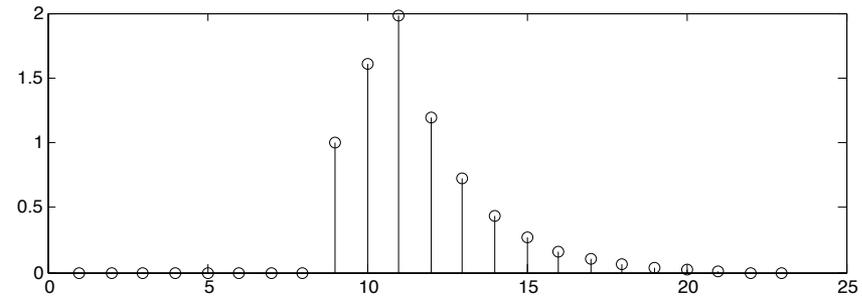
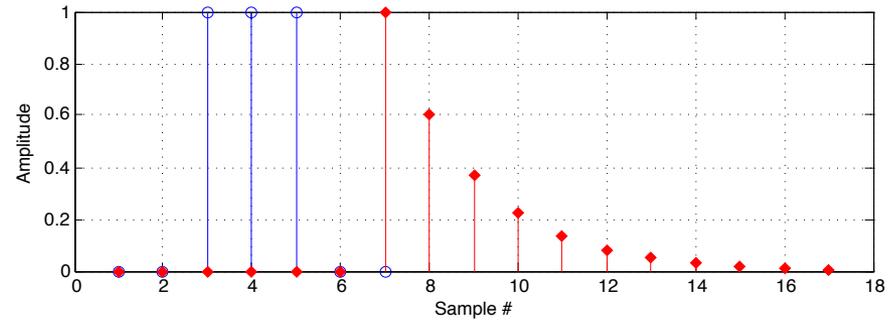


→ From a computational point of view, a relatively simple operation

```

wf1=[0 0 1 1 1 0 0];
% y= exp(-5*x) for x=[0,1]
wf2= [0 0 0 0 0 0 1.0000 0.6065 0.3679 0.2231 0.1353 0.0821 0.0498 0.0302 0.0183 0.0111 0.0067];
wf2= [0 0 0 0 0 0 0 0 1 1 1 0 0 0 0];
C= convolve1(wf1,wf2);

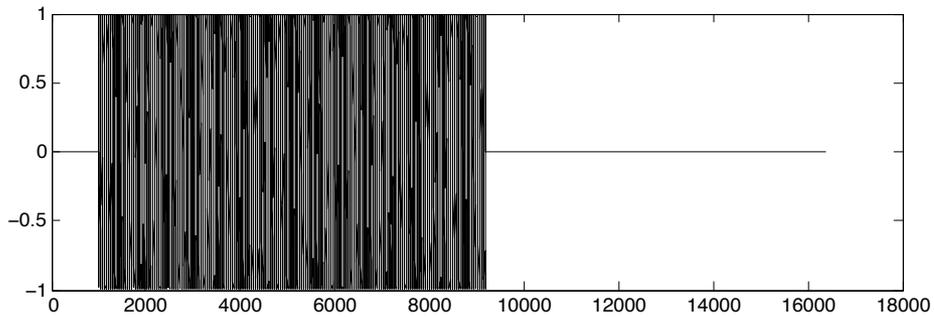
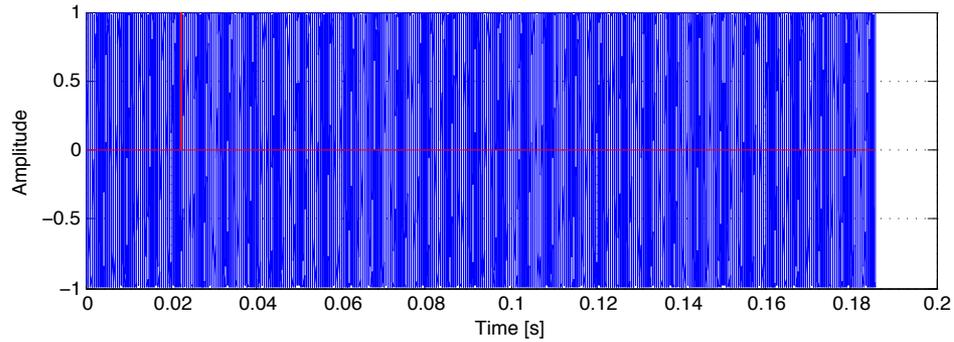
```



```

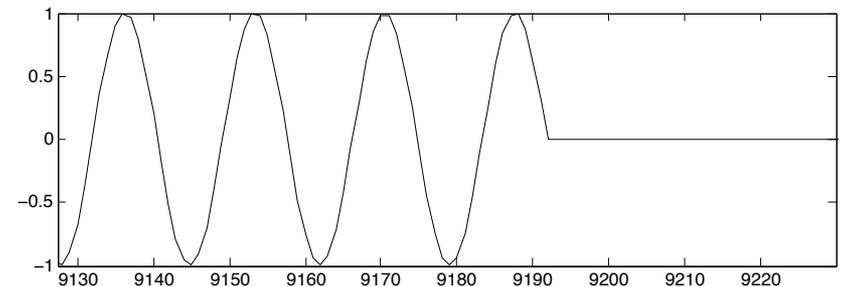
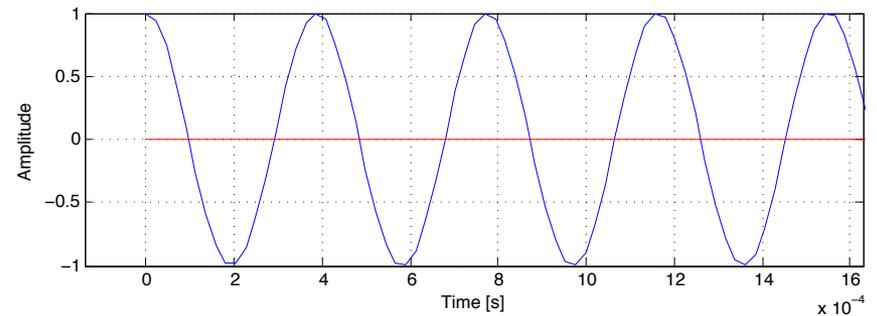
% ### EXconvolution2.m ###
% Example code to perform convolution between a sinusoid and narrow digital pulse
clear
% -----
SR= 44100;           % sample rate [Hz]
Npoints= 8192;      % length of fft window (# of points)
f= 2580.0;          % wf1: Frequency (for waveforms w/ tones) [Hz]
CLKbnd= [1000 1001]; % wf2: indicies at which pulse turns 'on' and then 'off'
% -----
% +++
t=[0:1/SR:(Npoints-1)/SR]; % create an array of time points
% +++
% create two waveforms (same dimensions)
wf1= cos(2*pi*f*t);
clktemp1= zeros(1,Npoints);
clktemp2= ones(1,CLKbnd(2)-CLKbnd(1));
wf2= [clktemp1(1:CLKbnd(1)-1) clktemp2 clktemp1(CLKbnd(2):end)];
% +++
% Use custom code (convolve1.m) or Matlab's built-in function? [should return identical
answers]
if 1==1
    C= convolve1(wf1,wf2); % custom code
else
    C= conv(wf1,wf2); % Matlab's built-in function
end
% +++
figure(1); clf;
subplot(211)
plot(t,wf1,'b'); hold on; grid on;
plot(t,wf2,'r'); xlabel('Time [s]'); ylabel('Amplitude');
subplot(212)
plot(C,'k'); hold on;

```



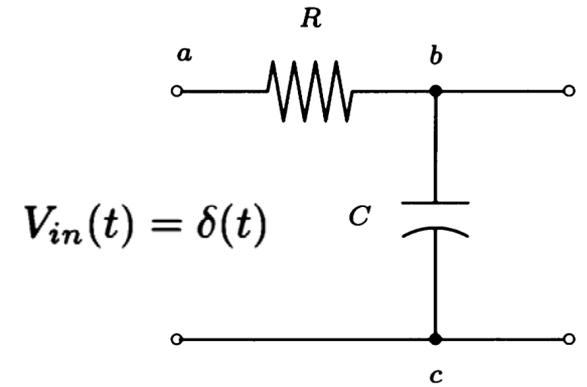
→ Interestingly, you seem to get back something that looks like (a slightly modified version of) the original.....
[this relates back to why we sample with pulses!]

Zoomed-in version



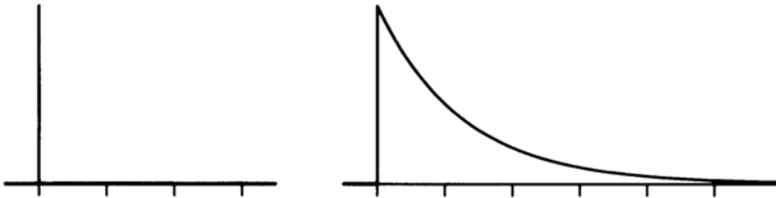
Connecting back to the electric circuit

- What is the connection between the circuit and the convolution?
- Imagine that the input is a string of impulses. Since the system is linear, the output will just be a sum of all the 'individual' responses

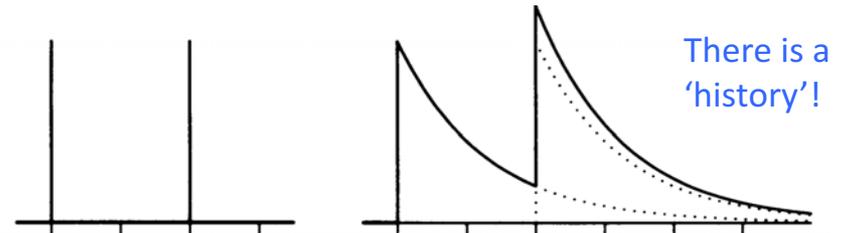


$$V_{out}(t) = \begin{cases} 0, & t < 0 \\ \frac{1}{RC} e^{-t/RC}, & t \geq 0 \end{cases}$$

Single impulse



Two successive impulses



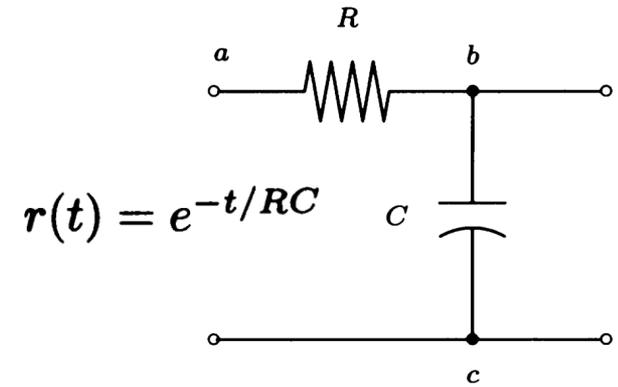
- We can imagine a limit with more and more impulses arriving at shorter and shorter interval, thereby passing to a continuous case

$$f(t) = \int_{-\infty}^{\infty} f(\tau) \delta(t - \tau) d\tau.$$

Here we have just written a (arbitrary) continuous function as an integral over delta functions

Connecting back to the electric circuit

- Let $r(t)$ represent the ‘**impulse response**’ of the system (i.e., the response to a delta function input)



- Writing our input voltage as... $V_{in}(t) = \int_{-\infty}^{\infty} V_{in}(\tau)\delta(t - \tau) d\tau$

- ... leads to an expression for the output as:
$$V_{out}(t) = \int_{-\infty}^{\infty} V_{in}(\tau)r(t - \tau) d\tau$$
$$= V_{in} \otimes r.$$

→ That is, the output of the system is the convolution of the input signal and the system’s impulse response

Put more generally, the impulse response totally characterizes the (linear) system!

- Remarkably, all this ties very nicely/importantly back to *Fourier transforms*....

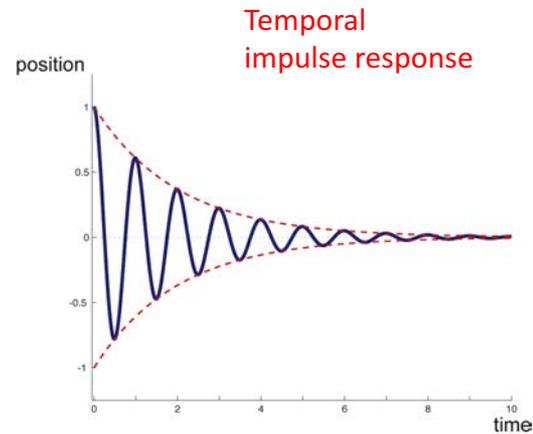
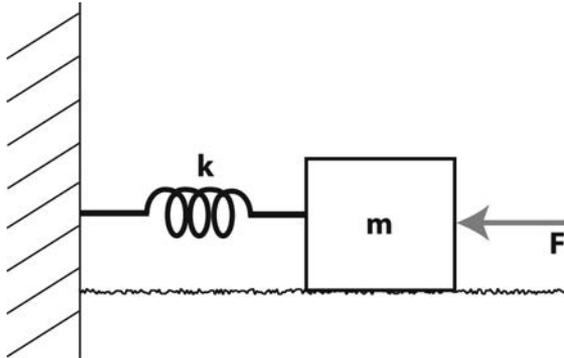
[we’ll come back to that in a bit]

Impulse response

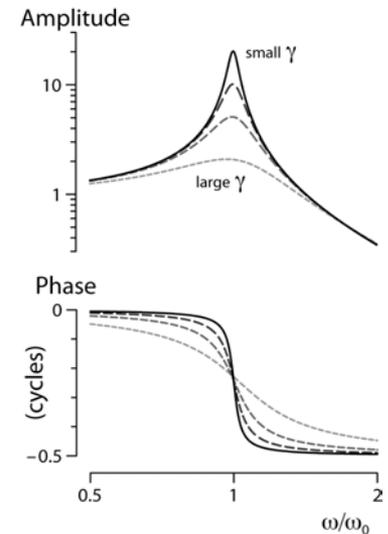
➤ Intuitively defined in two different (but equivalent) ways:

1. Time response of 'system' when subjected to an impulse
(e.g., striking a bell w/ a hammer)
2. Fourier transform of resulting response
(e.g., spectrum of bell ringing)

ex. Harmonic oscillator



Spectral
impulse response

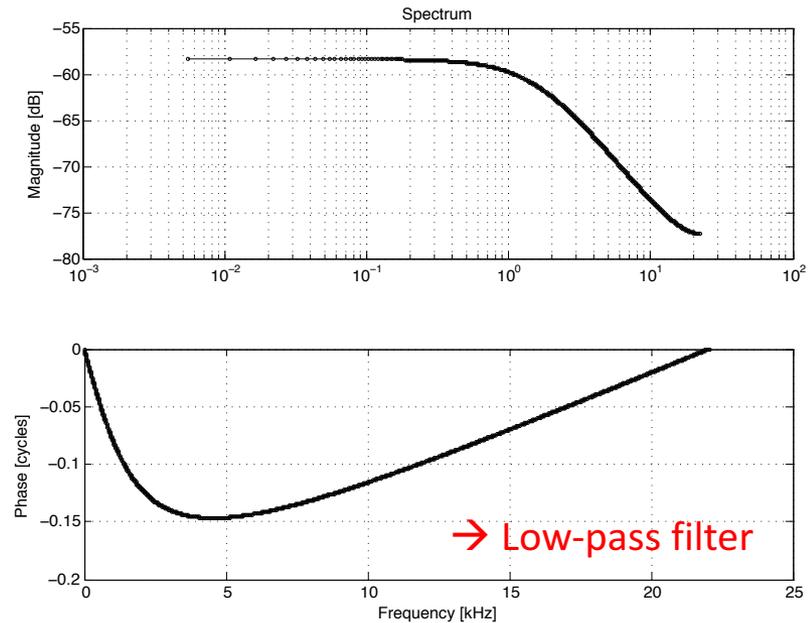
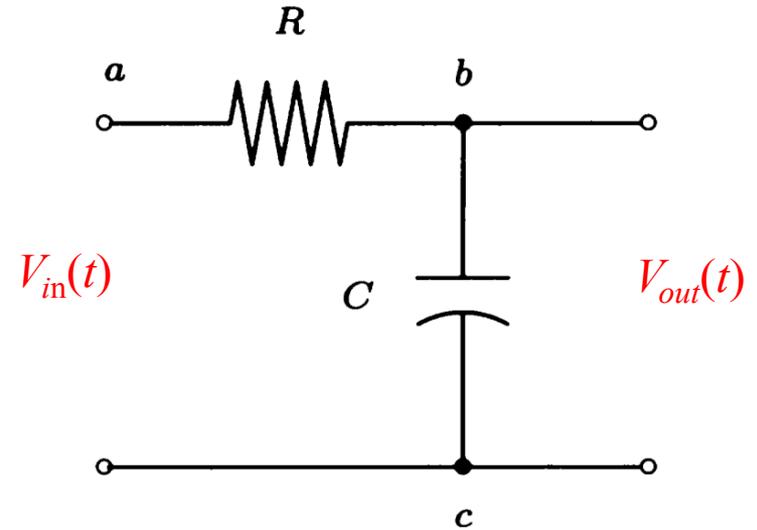
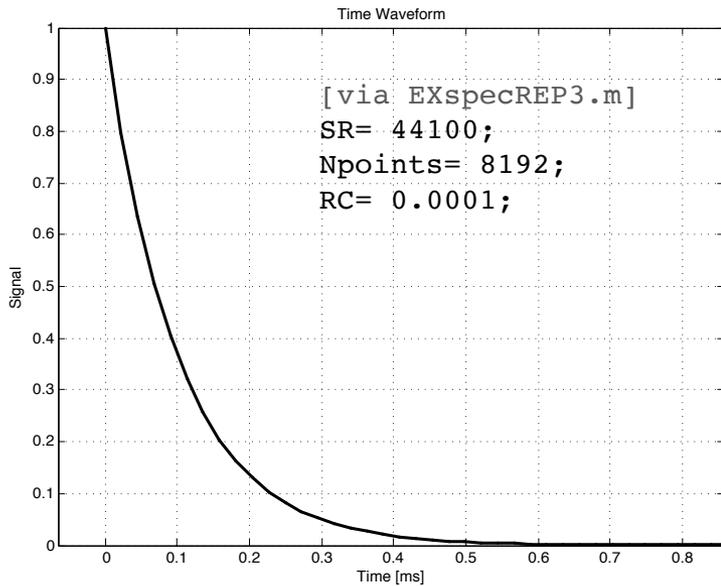


(Important) Note: The Fourier transform of the impulse response is called the *transfer function*

Ex. Electric circuit's 'impulse response'

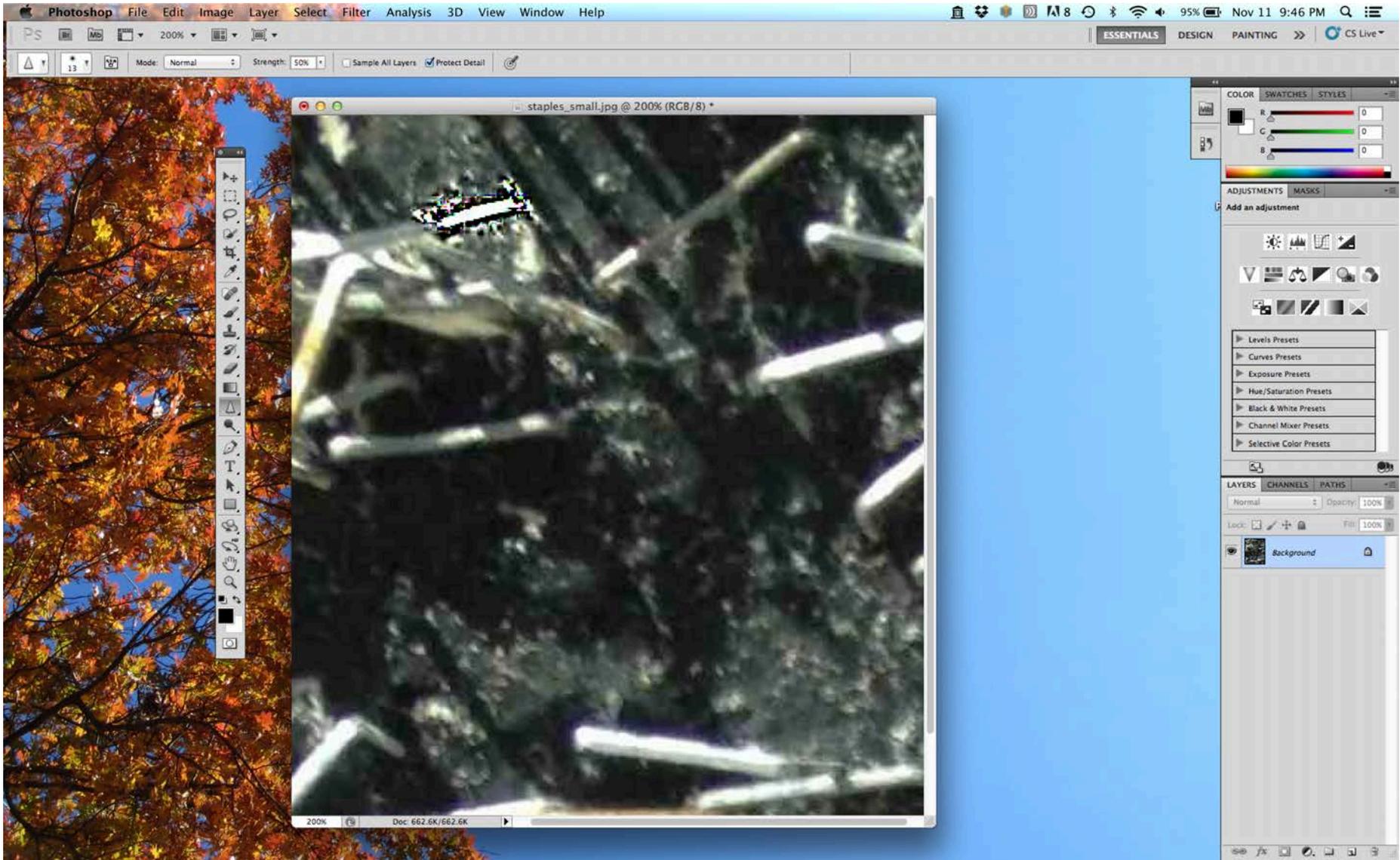
- How does this all tie back (intuitively) to our original circuit problem?
- Fourier transform of the impulse response 'attenuates' at high frequencies

$$r(t) = e^{-t/RC}$$



→ This introduces an important topic known as 'filtering'

Ex. Photoshop & 'sharpening'



→ How does the 'sharpen' tool (or Sharpen 'filter') work in Photoshop?

2-D Convolutions: Images & 'Filtering'

Case Study: Microscope Imaging

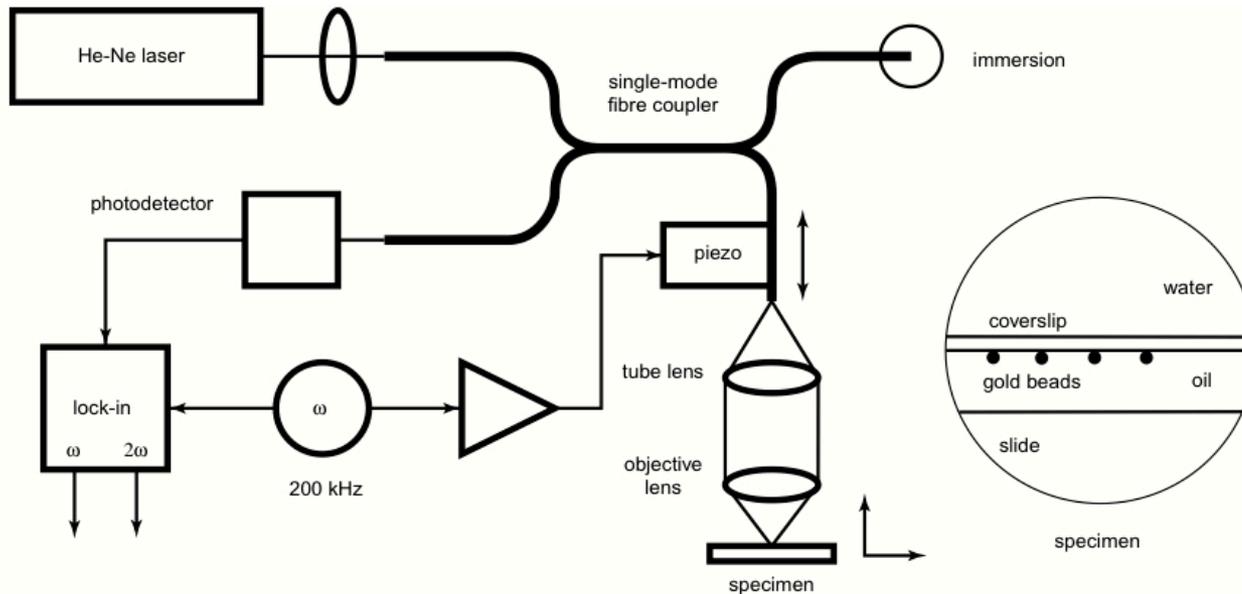


FIGURE 11.2. Schematic diagram of the fiber-optic interferometer-based setup for measuring objective PSFs.

➤ Light takes a complex path through the optics

➤ Factor in the wave nature of light, things get somewhat complicated....

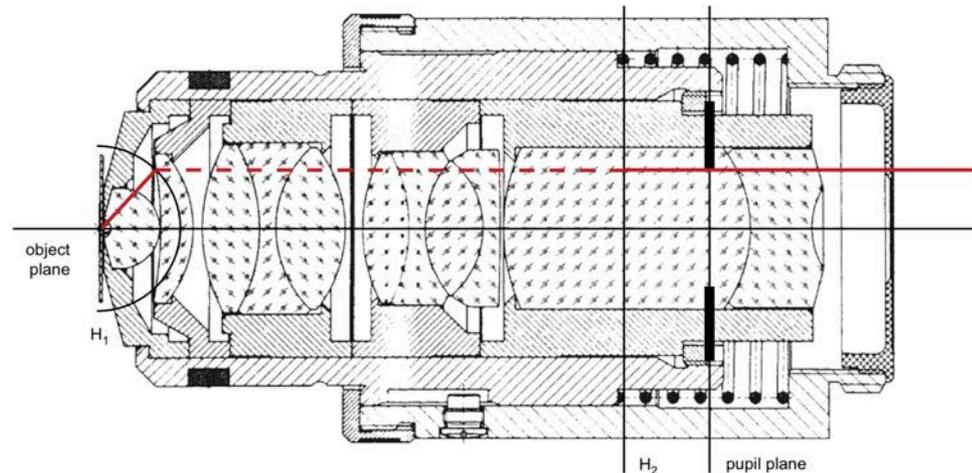
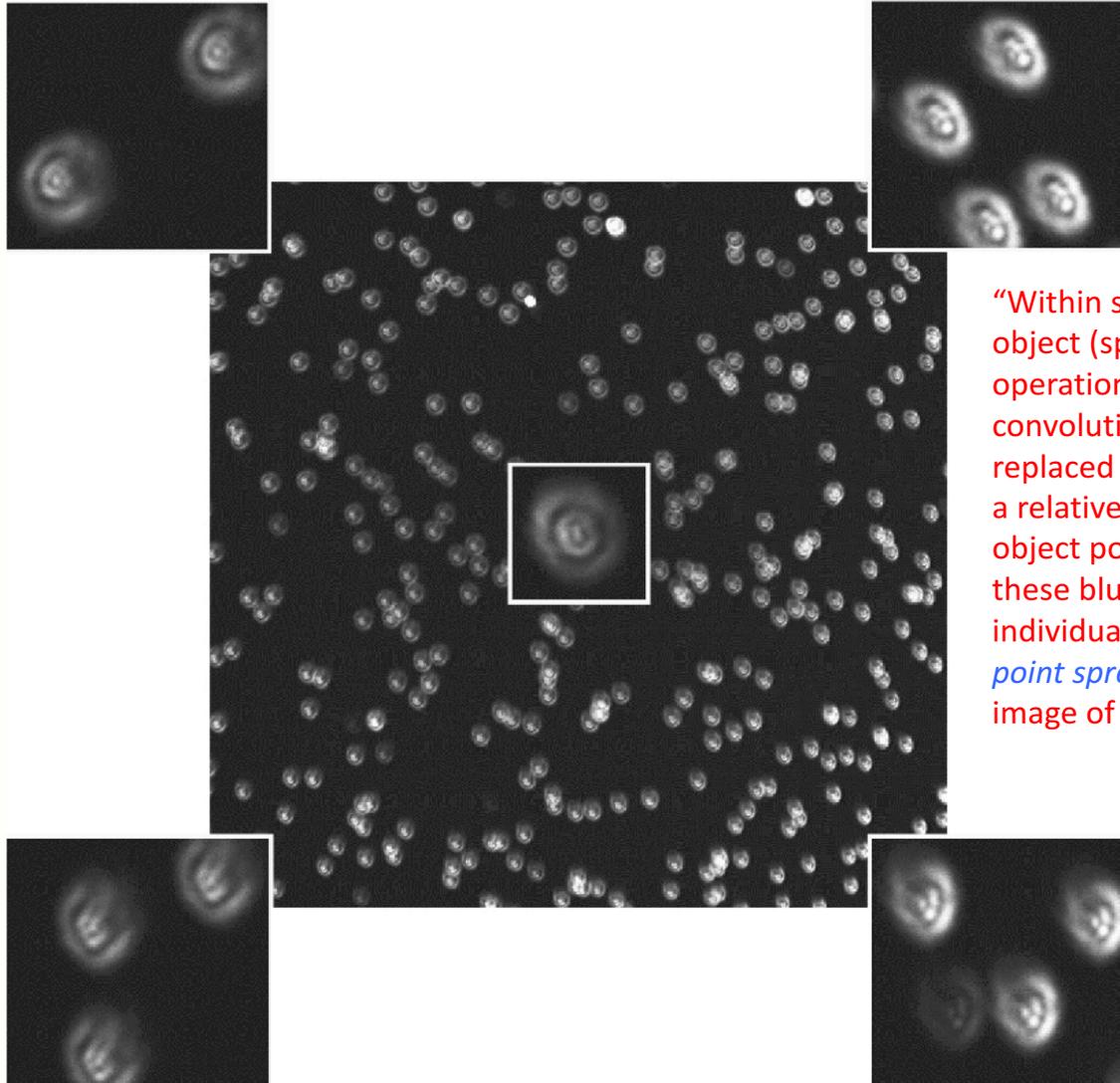


FIGURE 11.1. Schematic diagram of a typical high NA planapochromat objective lens. Principal surfaces, aperture stop, and marginal ray are indicated.



“Within some quite general limitations, the object (specimen) and image are related by an operation known as *convolution*. In a convolution, each point of the object is replaced by a blurred image of the point having a relative brightness proportional to that of the object point. The final image is the sum of all these blurred point images. The way each individual point is blurred is described by the *point spread function* (PSF), which is simply the image of a single point.”

FIGURE 23.5. Confocal image of a field of beads. The focal plane has been set away from the central, in-focus, section to show some of the out-of-focus parts of the PSF. The lack of shift invariance is evident in this image, as the defocused bead structure is different in different parts of the field of view. The insets show enlargements from different parts of the field. The image is also dimmer in the central area due to field curvature. (Objective used was Leitz Planapo, NA 1.4, oil immersion.)

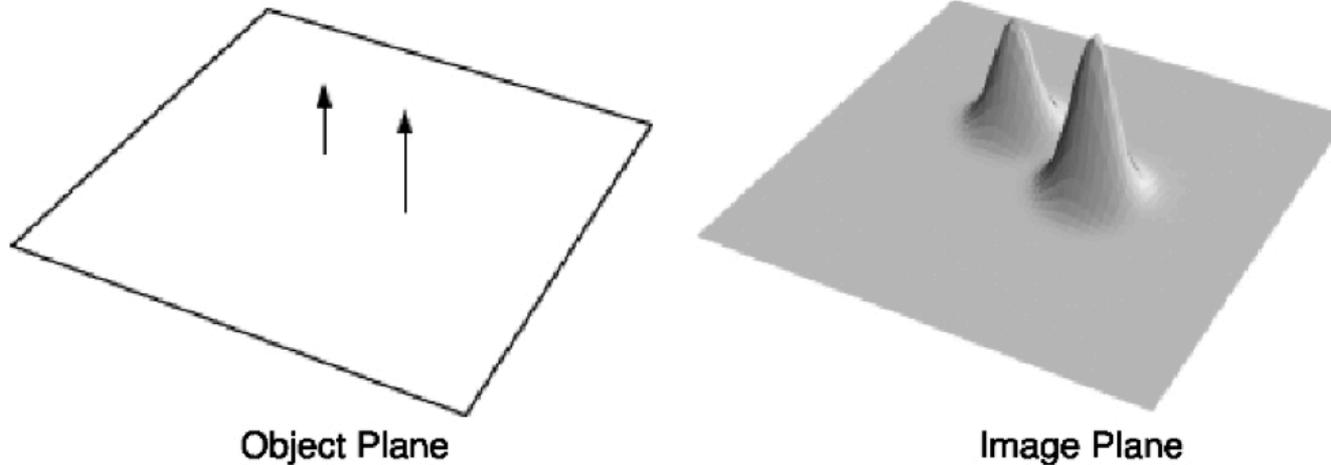
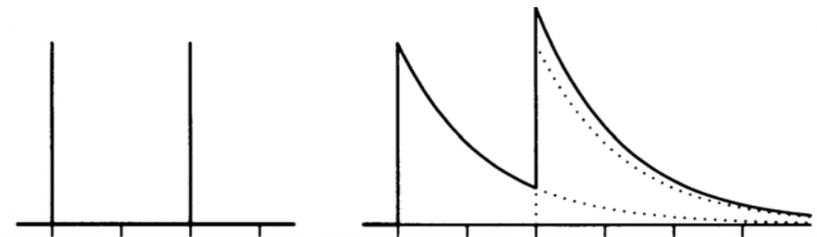


FIGURE 12.4. The point-spread function. Two impulse sources of different height are shown in the object plane. The response to them is shown in the image plane.

→ The PSF is directly akin to an 'impulse response'



Point Spread Function (PSF)

Case Study: Microscope Imaging

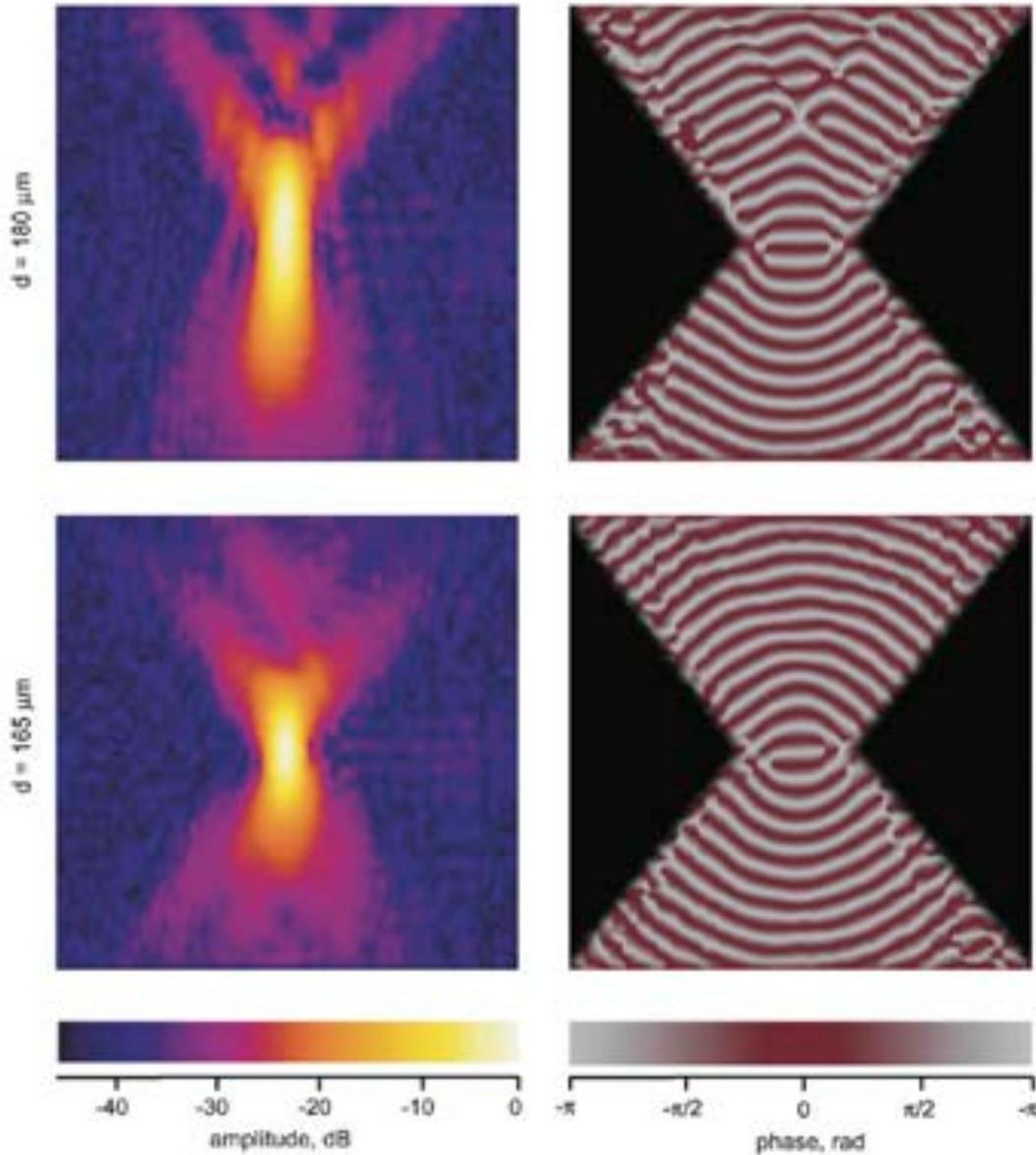


FIGURE 11.3. The amplitude and phase of the effective PSF for 60×1.2 NA water-immersion lens with correction collar. Results for two different collar settings are shown. Image size in both x (horizontal) and z (vertical) are $5 \mu\text{m}$.

How to measure PSF?

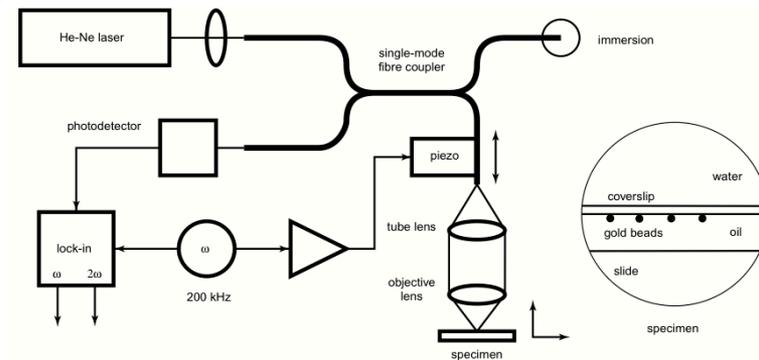


FIGURE 11.2. Schematic diagram of the fiber-optic interferometer-based setup for measuring objective PSFs.

2-D Convolutions: Images & 'Filtering'

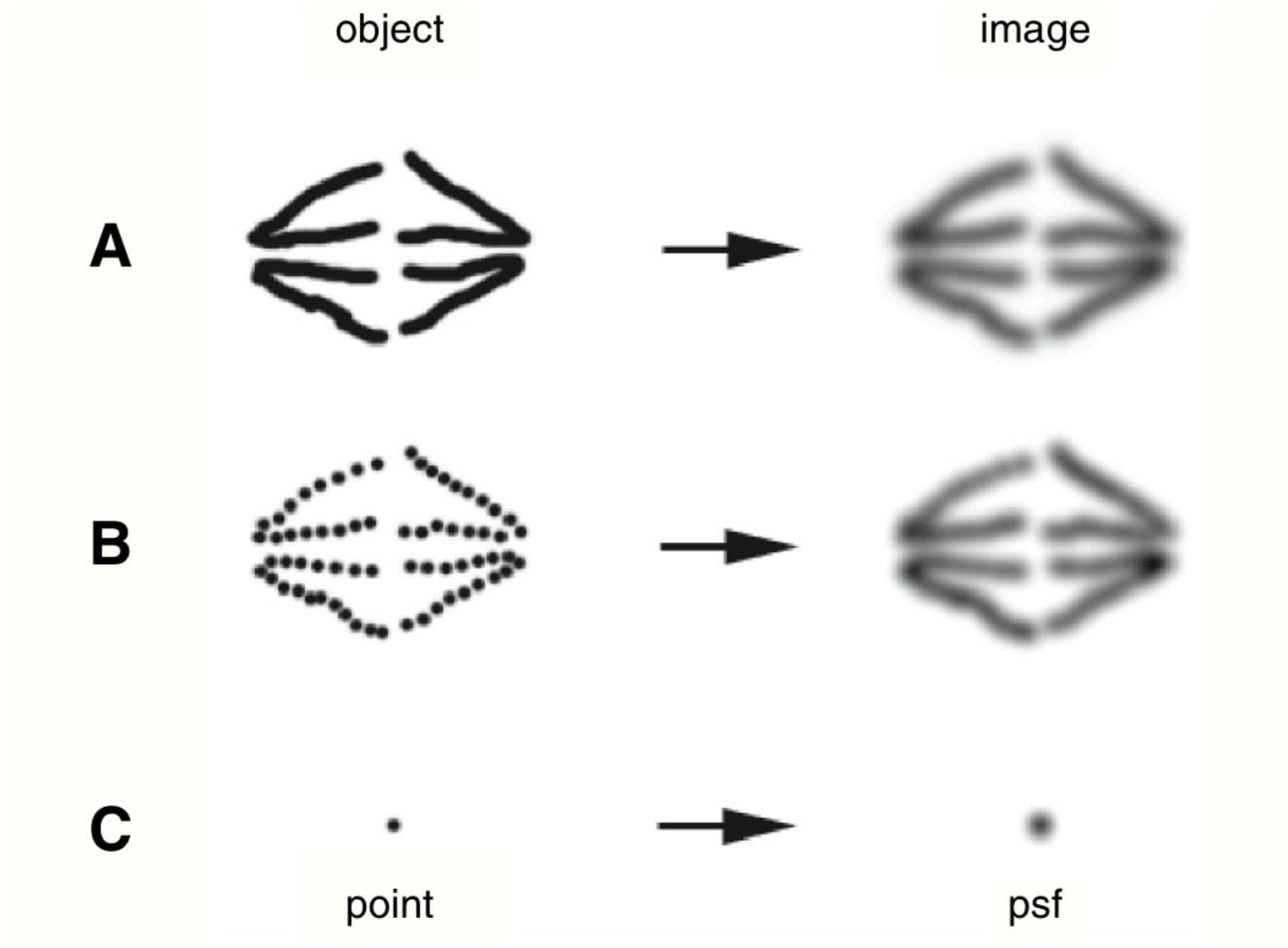
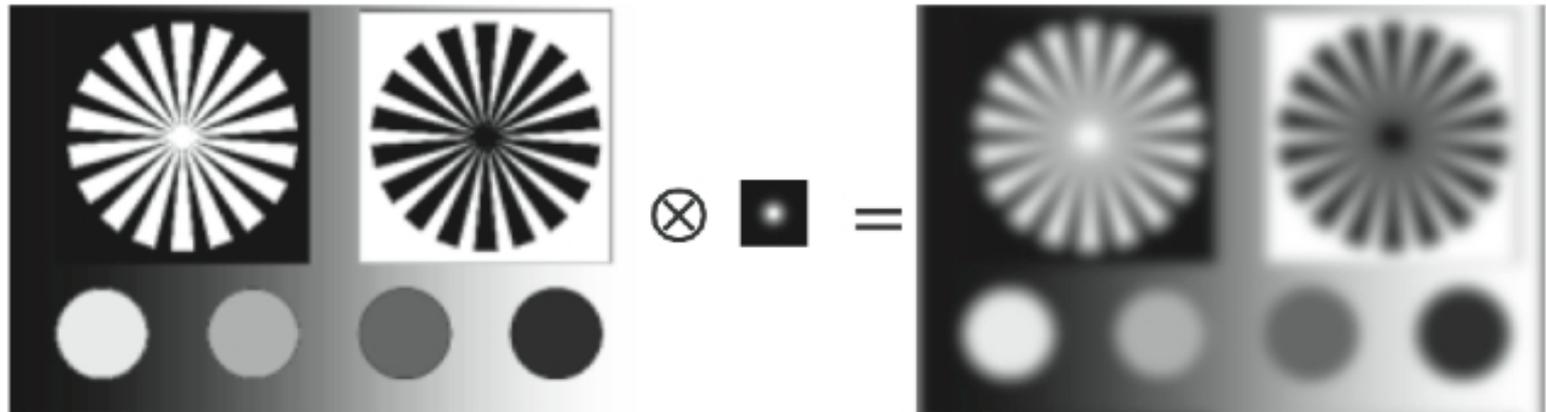
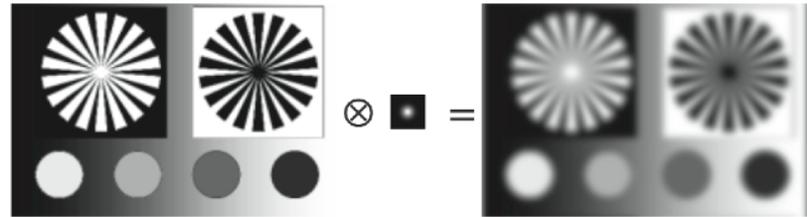


FIGURE 23.1. Diagram showing how a single point is imaged as the PSF by a microscope, and thus that the image of an extended object is the convolution of the object with the PSF.

2-D Convolutions: Images & 'Filtering'





Convolution → Blurring

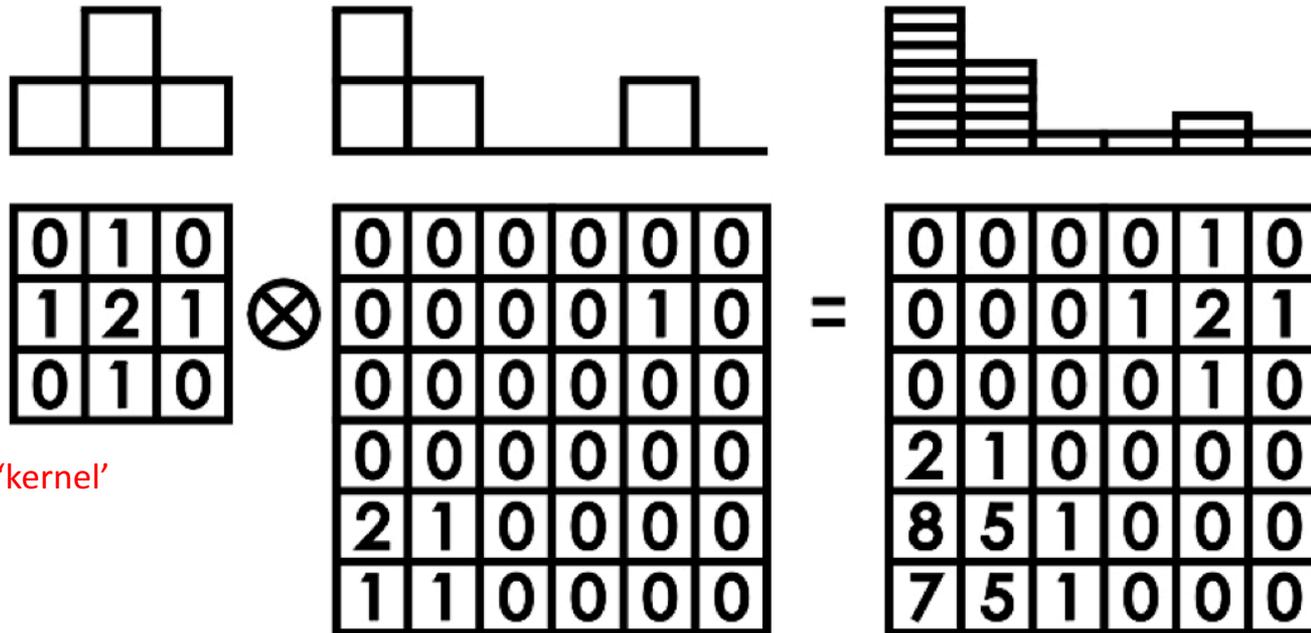
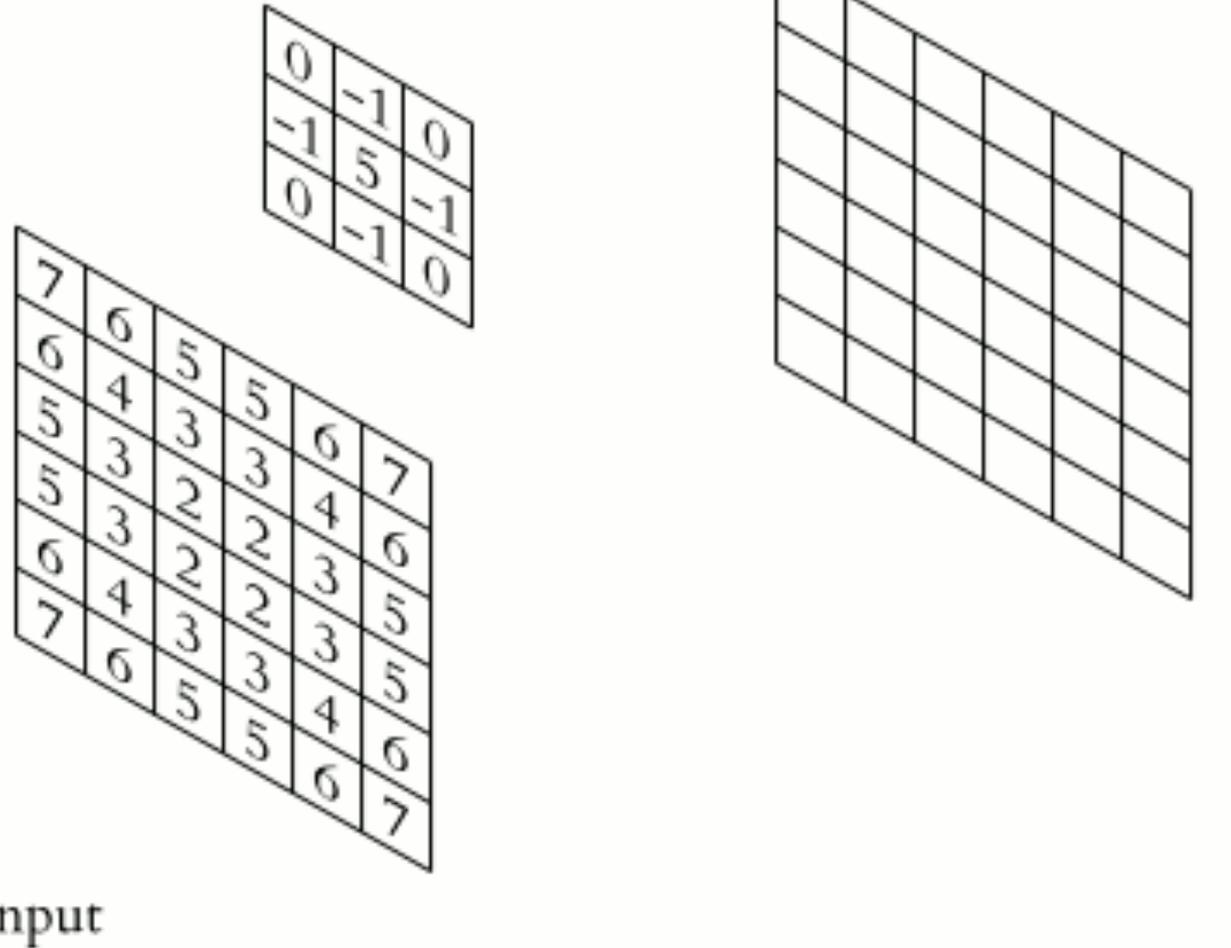


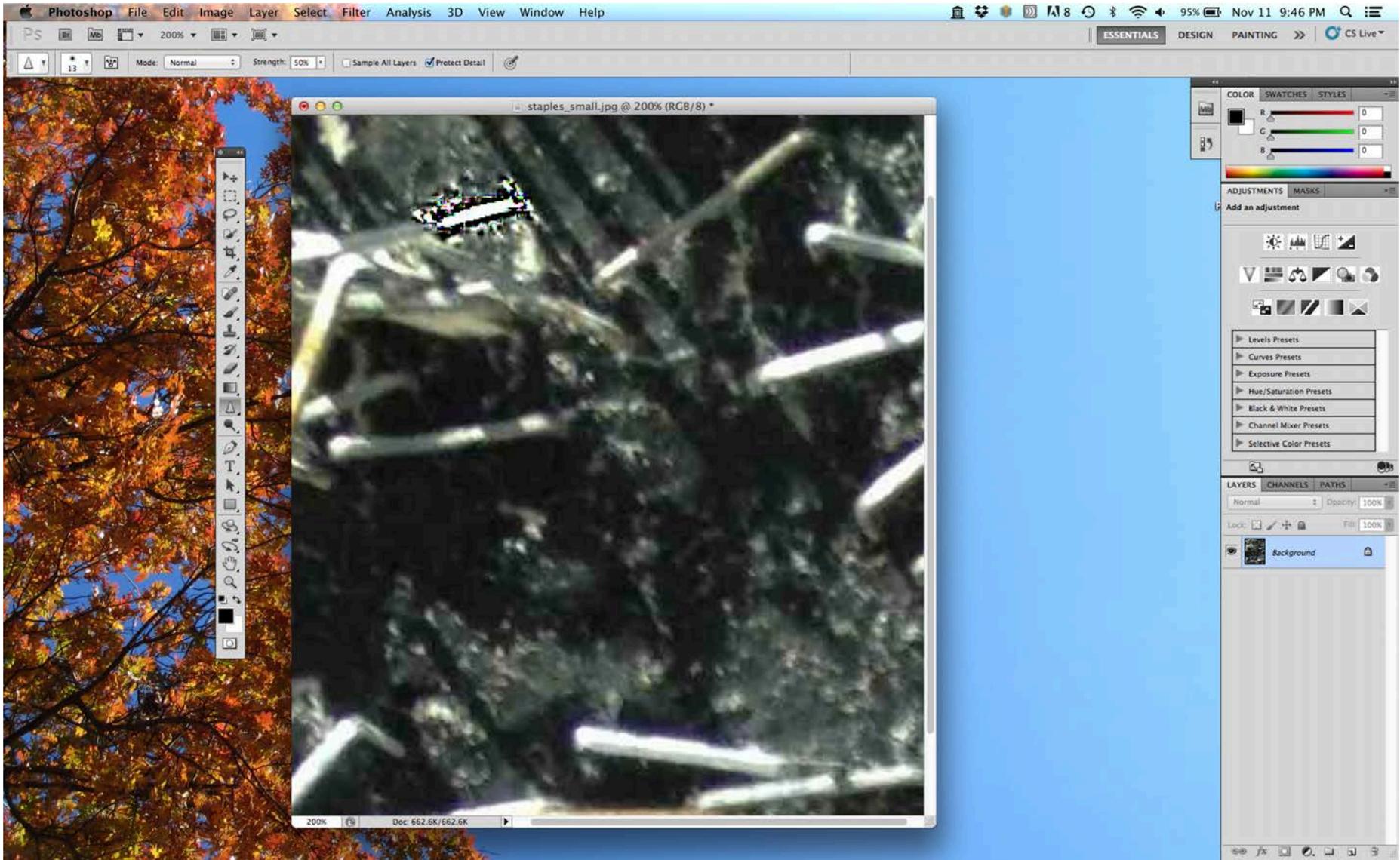
FIGURE 25.2. Schematic diagram demonstrating the convolution (\otimes) operation with a 6×6 pixel object and a 3×3 pixel blurring kernel. The profiles above show the maximum projection of the two-dimensional grids as would be seen looking across the planes from above. Note how the contrast of the peaks in the image is reduced and smeared across the image.

Image Processing: Filtering

Convolution → Sharpening



Basic idea starts to provide some intuition for how image processing (e.g., Photoshop) works!



→ Photoshop uses a specific type of 'kernel' and convolves in the spatial domain

```

function EXsharpenImage(file,type);          % A. Salerno [BPHS 4090 F13]
% For specified image, a user-defined kernel (see below) is convolved with
% the image in the spatial domain. The two default kernels are a blurring
% and a sharpening [see also http://en.wikipedia.org/wiki/Kernel_(image_processing) ]
% ex. > EXsharpenImage('filename.png','s');
close all
% specify kernel to be used
if nargin<2
    type = 's';% Sharpening is default
end
if strcmp(type,'s') || strcmp(type,'sharp')
    ker = [-1 -1 -1
           -1 50 -1
           -1 -1 -1];
elseif strcmp(type,'b') || strcmp(type,'blur')
    ker = [1 1 1
           1 1 1
           1 1 1]; %Adaptation of blurring kernel on wiki
elseif strcmp(type,'e') || strcmp(type,'edge')
    ker = [-1 -1 -1
           -1 8 -1
           -1 -1 -1]; % edge detection
end

im = imread(file); % Imports an image
if numel(size(im)) == 3 %Only for rgb images
    im = rgb2gray(im); %converts the image into a grayscale
end

%Plot original Image in b/w
figure
subplot(1,2,1)
imagesc(im); colormap(gray); colorbar;
axis image; title('Original Image');
n = size(im);
imp = zeros(n(1)+2,n(2)+2); % imp is the matrix that will be used
% to process the data. It is the "extended matrix" used.
% The addition of 2 is for the border (i.e.
% there is an addition at the beginning and at
% the end)
imp(2:end-1,2:end-1) = im; % Set the centre of the processing matrix to
% be the original matrix - basis for how the
% image will be processed

%Set the corner values to be the same
imp(1,1) = im(1,1);
imp(end,1) = im(end,1);
imp(1,end) = im(1,end);
imp(end,end) = im(end,end);

% Set the centred "edge rows and columns" to be the
same
imp(1,2:end-1) = im(1,:); % Top row
imp(end,2:end-1) = im(end,:); %Bottom row
imp(2:end-1,1) = im(:,1); %Left Column
imp(2:end-1,end) = im(:,end); %Right column

% Since the matrix can be scaled in multiple different
ways, the top corner
% seems the easiest to do. Using the top corner of
imp, I will scan across
% imp, multiplying the elements in 3x3 matrices,
summing them, and then
% taking that value, and putting into 'fim' the Final
Image
im2 = zeros(n(1),n(2));
for i = 1:n(1) % Rows
    for j = 1:n(2) %columns
        var = ker.*imp(i:i+2,j:j+2);
        im2(i,j) = sum(sum(var));
    end
end

% In order to prevent errors, I will now normalize the
matrix to it's
% greatest value, set this to be the max of the
original matrix ensuring
% every value ends up as an integer
subplot(1,2,2)
imagesc(im2); colorbar;
axis image; title('Image with Kernel Applied')

```

Connection to Fourier Transforms

- Consider the Fourier transform of the (1-D) convolution:

$$\begin{aligned}\mathcal{F}[p \otimes q] &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} [p \otimes q] e^{-i\omega t} dt \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \left[\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} p(\tau) q(t - \tau) d\tau \right] e^{-i\omega t} dt \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} p(\tau) \left[\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} q(t - \tau) e^{-i\omega t} dt \right] d\tau.\end{aligned}$$

- Making use of the 'shifting property', the term in the square brackets is:

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} q(t - \tau) e^{-i\omega t} dt = e^{-i\omega\tau} Q(\omega)$$

$Q(\omega)$ is the Fourier transform of $q(t)$

$$\begin{aligned}\mathcal{F}[p \otimes q] &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} p(\tau) e^{-i\omega\tau} Q(\omega) d\tau \\ &= P(\omega) Q(\omega),\end{aligned}$$

Convolution theorem

$$\mathcal{F}[p \otimes q] = \mathcal{F}[p] \mathcal{F}[q]$$

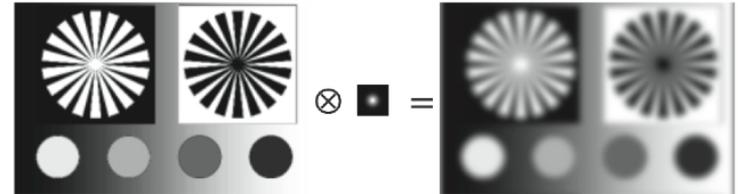
$P(\omega)$ is the Fourier transform of $p(t)$

Convolution Theorem

- Simple but powerful idea:
Convolution in the time domain is simply a multiplication in the spectral domain

$$\mathcal{F}[p \otimes q] = \mathcal{F}[p] \mathcal{F}[q]$$

- Door swings both ways: From the output, if we know the impulse response, we can *deconvolve* (i.e., divide in spectral domain) to get the original input!



$$V_{out} = V_{in} \otimes r$$

$$\mathcal{F}[V_{out}] = \mathcal{F}[V_{in} \otimes r] = \mathcal{F}[V_{in}] \mathcal{F}[r]$$

$$\mathcal{F}[V_{in}] = \frac{\mathcal{F}[V_{out}]}{\mathcal{F}[r]}$$

$$V_{in}(t) = \mathcal{F}^{-1} \left[\frac{\mathcal{F}[V_{out}]}{\mathcal{F}[r]} \right]$$

Ex. Deconvolution & Image Processing

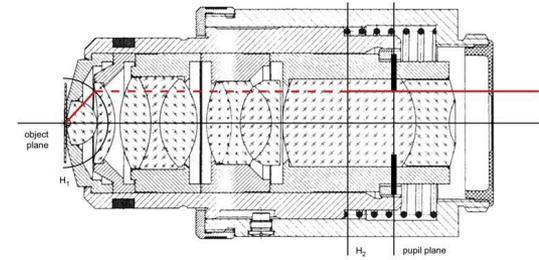
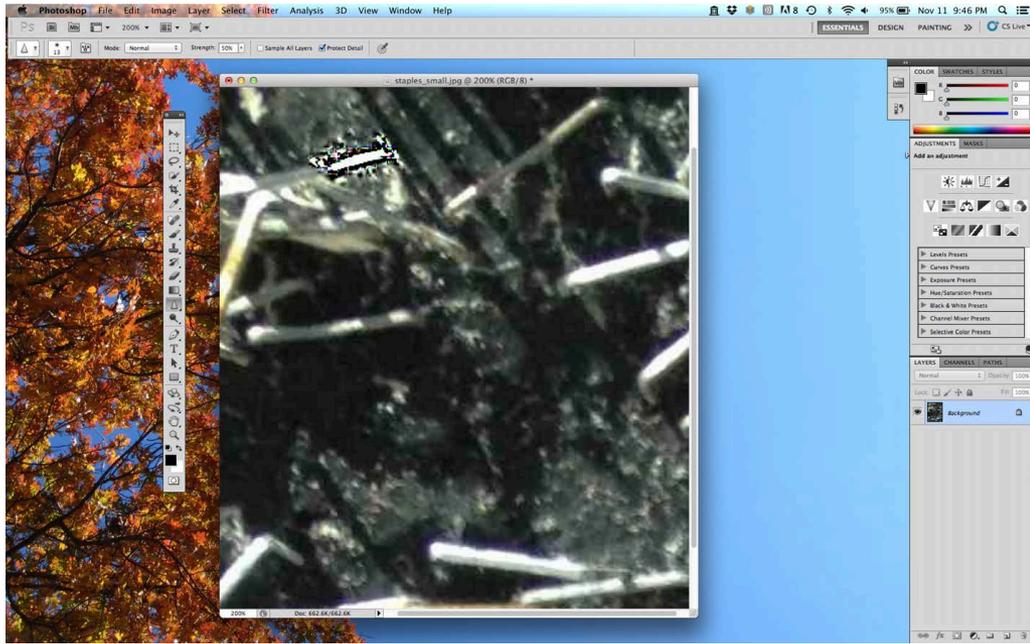


FIGURE 11.1. Schematic diagram of a typical high NA planapochromat objective lens. Principal surfaces, aperture stop, and marginal ray are indicated.

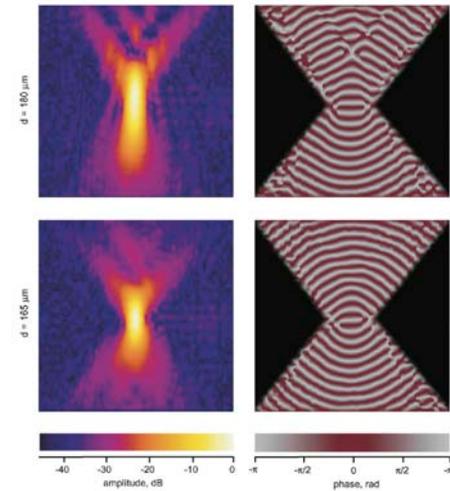
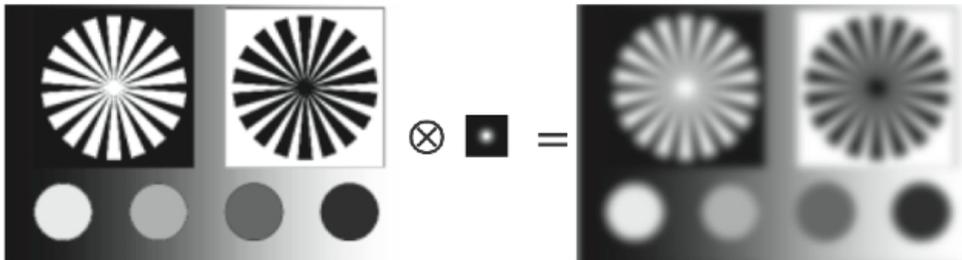
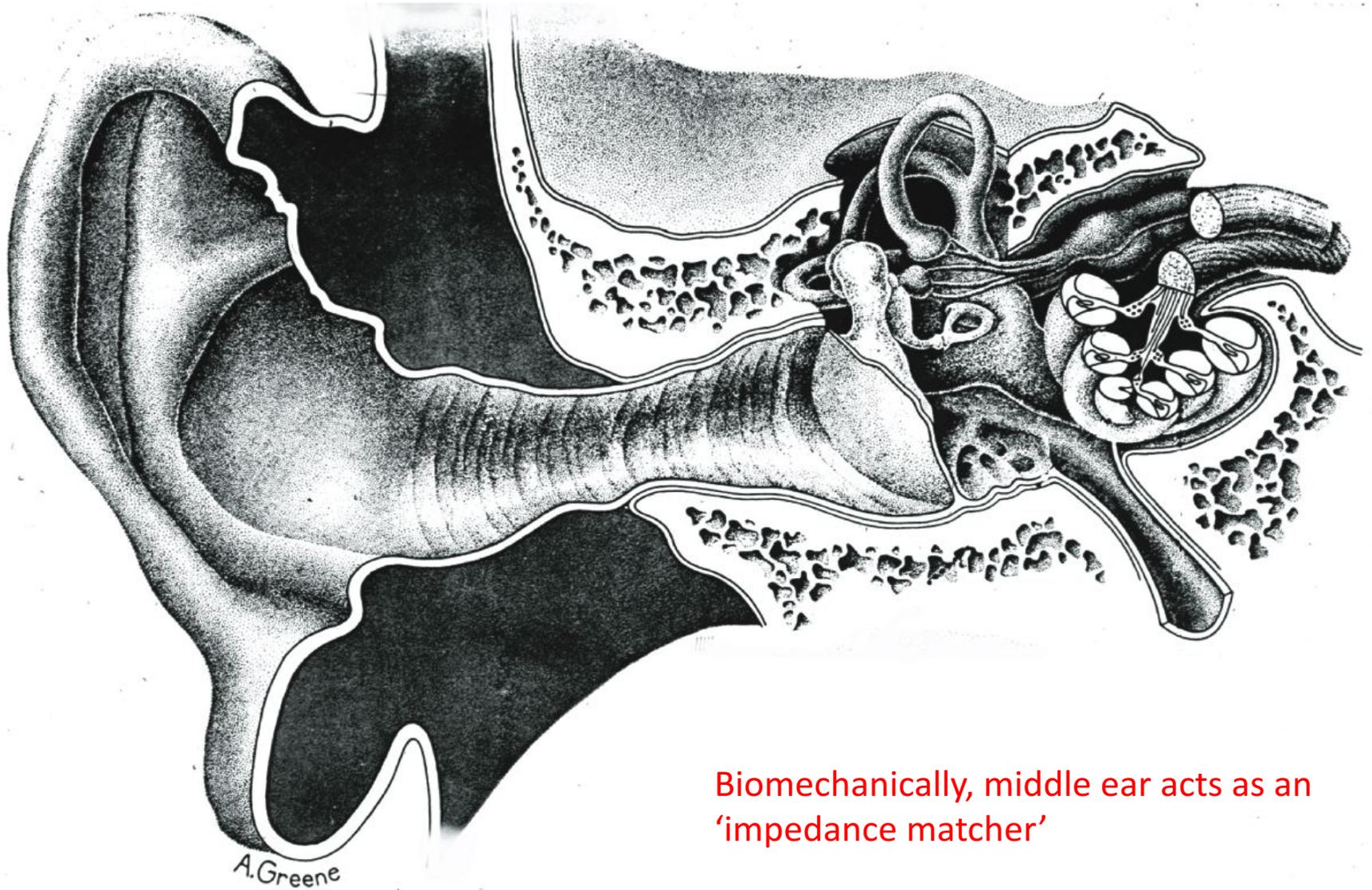


FIGURE 11.3. The amplitude and phase of the effective PSF for 60×1.2 NA water-immersion lens with correction collar. Results for two different collar settings are shown. Image size in both x (horizontal) and z (vertical) are $5 \mu\text{m}$.



→ If you have some estimate of your PSF, you can 'correct' your image very efficiently in the frequency domain

Ex. Middle ear



Biomechanically, middle ear acts as an 'impedance matcher'

Ex. Middle ear

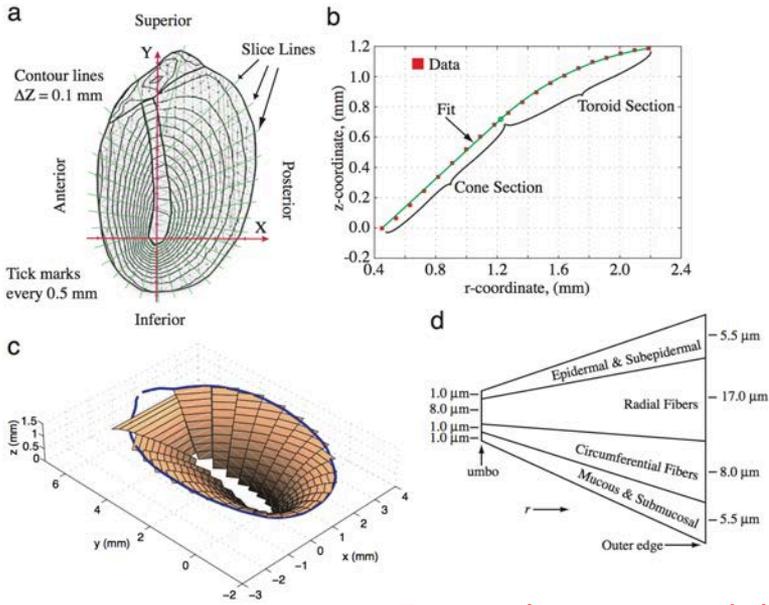
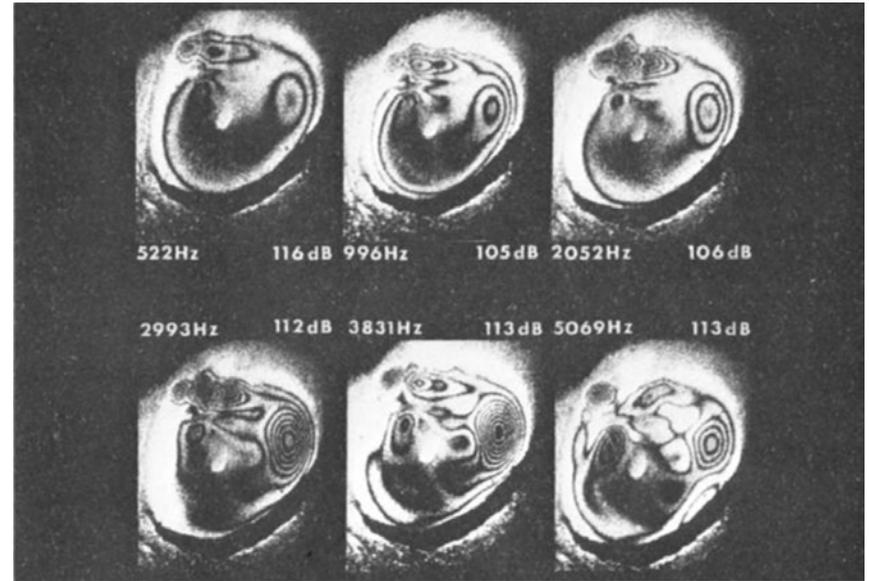
Tonndorf & Khanna (1972)

de la Rouchfoucauld & Olson (2010)



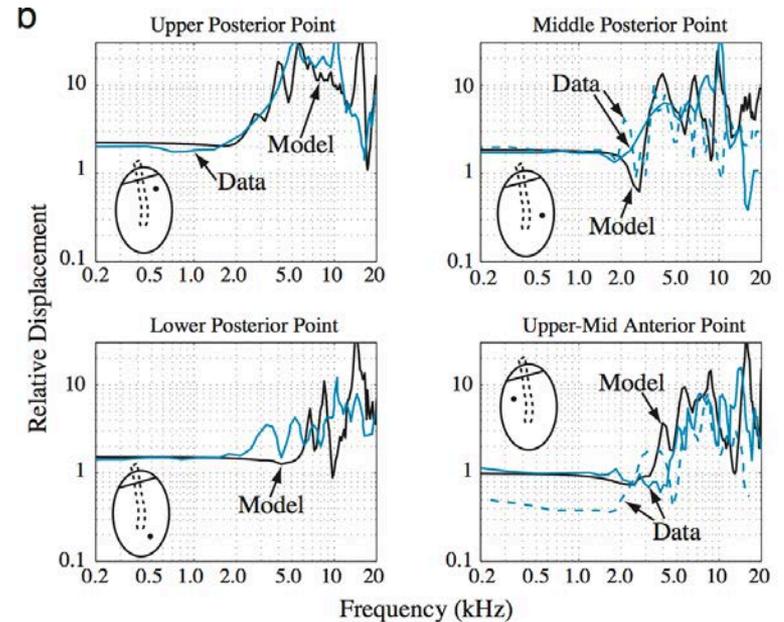
➤ Eardrum is a thin, tent-like membrane

➤ Sound-induced motion is surprisingly complex



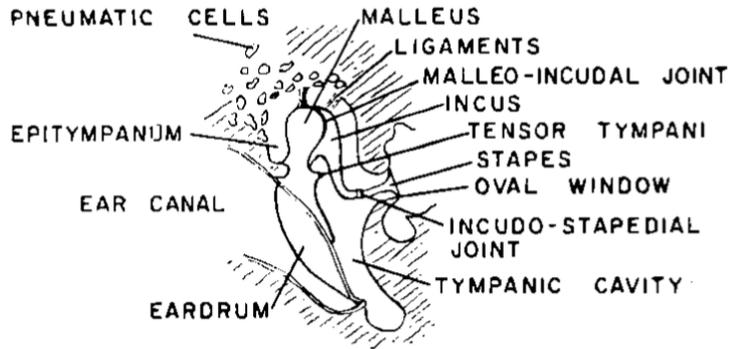
➤ Finite element models are one approach.... (but require lots of assumptions about parameters)

Fay et al. (2006)



Ex. Middle ear

- ... but a surprisingly effective approach is much simpler: *'lumped elements'*



(i.e.,
electrical circuit analog)

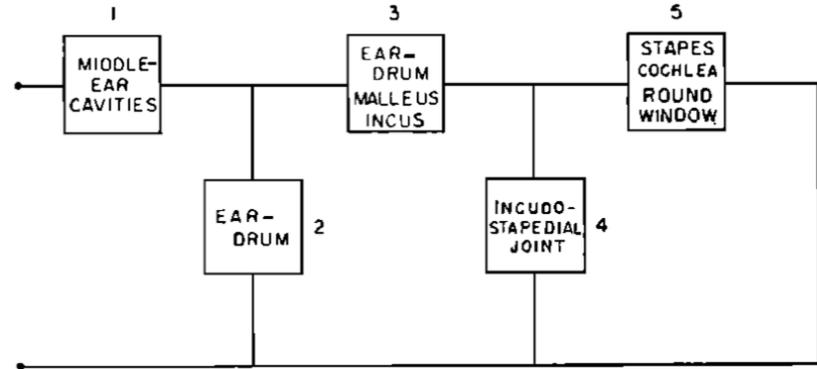
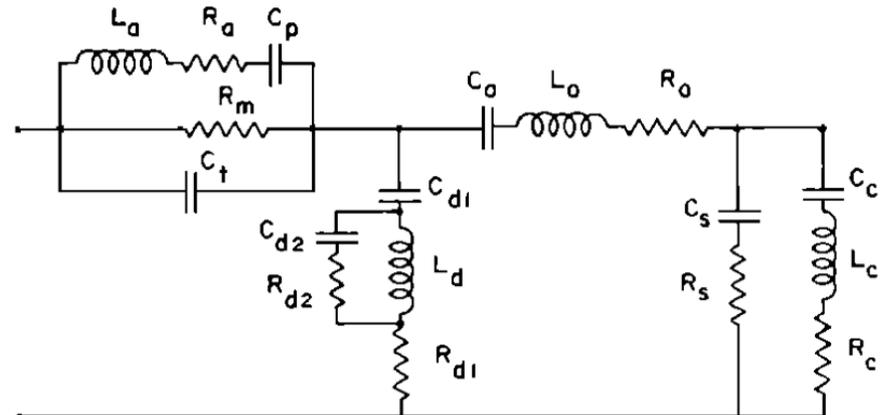


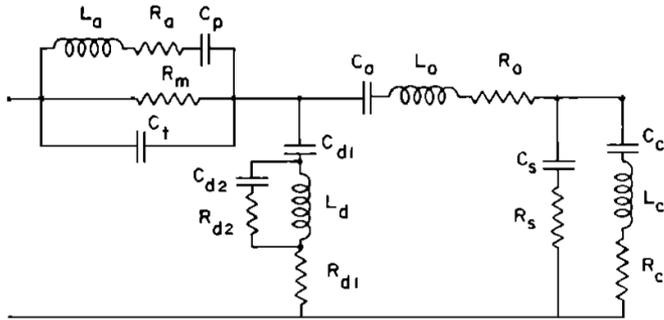
FIG. 2. Block diagram of the middle-ear mechanism.

$$C = V/\rho c^2$$

→ Various circuit elements are acoustic analogs (e.g., capacitance represents springiness of air compression of closed cavity)

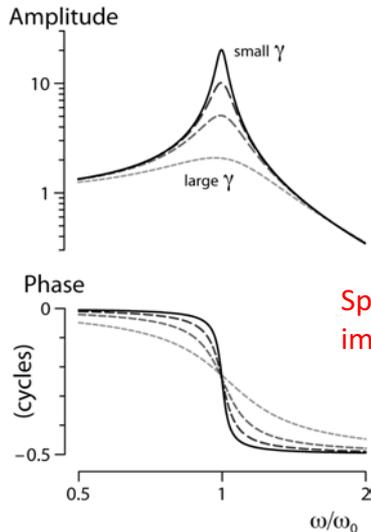
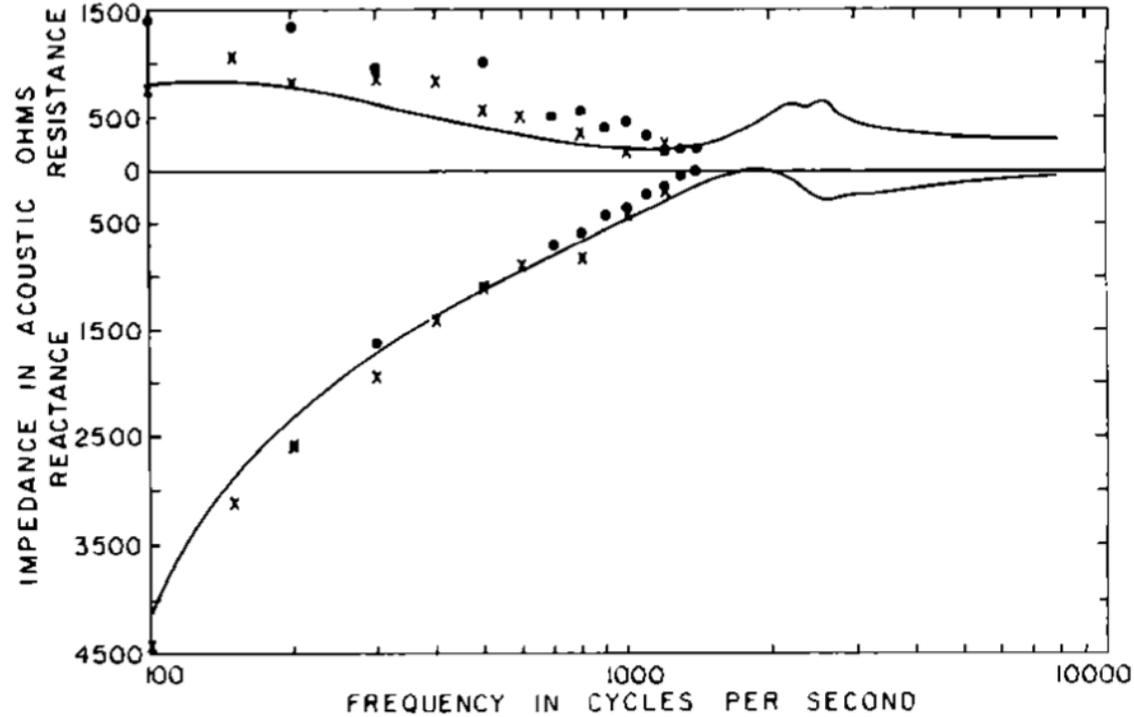


Ex. Middle ear



- To decent first order, model captures many experimental features of sound transmission

- Essentially amounts to the system's impulse response (or the closely related 'transfer function')



Spectral impulse response

$$V_{out}(t) = \int_{-\infty}^{\infty} V_{in}(\tau)r(t - \tau) d\tau = V_{in} \otimes r.$$

→ We've characterized the 'filtering' properties of the middle ear!

Post-class exercises

- What happens when the position of the resistor and capacitor are swapped?

- Confirm that the given solution does indeed satisfy the ODE.

$$\frac{dV_{out}}{dt} + \frac{V_{out}}{RC} = \frac{V_{in}}{RC}$$

$$V_{out}(t) = e^{-t/RC} \left[\int_{-\infty}^t e^{\tau/RC} V_{in}(\tau) d\tau + C_1 \right]$$

- If you wanted to build a low-pass filter with a particular ‘cutoff frequency’, what would you pick for R and C? How exactly might you construct the circuit (e.g., what gets soldered to what)?

- For image processing, what sort of kernel would you use for ‘edge detection’? How do you think this relates to a ‘mask’ in the spectral domain?

- Using `EXconvolve1.m`, try varying the waveforms and observe the different patterns that arise. Are they consistent with what you expected?

