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

York University
Winter 2018
Lecture 3

Review: Numerical differentiation

Derivative (by definition)

$$\frac{df(t)}{dt} = \lim_{\Delta t \to 0} \frac{f(t+\Delta t) - f(t)}{\Delta t}. \qquad (4.1.1)$$

Simple approximation of the first derivative

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

Taylor series expansion about x

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(\xi)$$

Note: This equality is not strictly true!

Note: We are making some assumptions about $f(x)$ (e.g., differentiability, continuity)

"Forward difference"

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2} f''(\xi)$$

Note: Only including lowest order terms from Taylor series (quadratic captures the "error" or the truncation term)

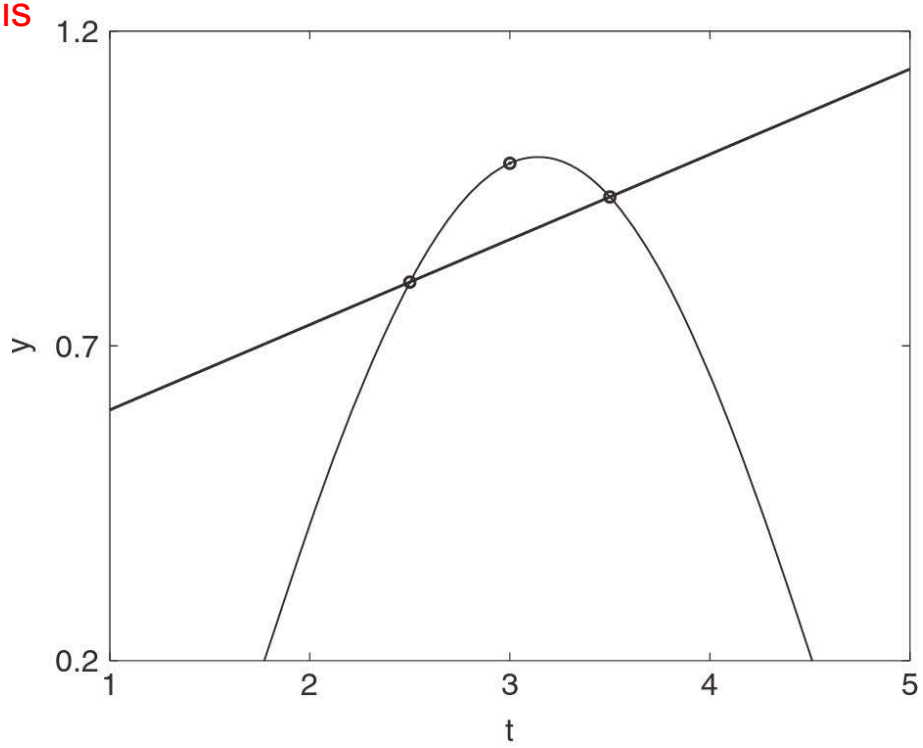
Kutz (2013)
http://www2.math.umd.edu/~dlevy/classes/amsc466/lecture-notes/differentiation-chap.pdf

Review: Numerical differentiation

$$\frac{df(t)}{dt} = \frac{f(t+\Delta t) - f(t-\Delta t)}{2\Delta t} - \frac{\Delta t^2}{6}\frac{d^3 f(c)}{dt^3}$$

Since time step is small, higher terms become relatively small (but not necessarily negligible!)

Note: This equality is not strictly true!



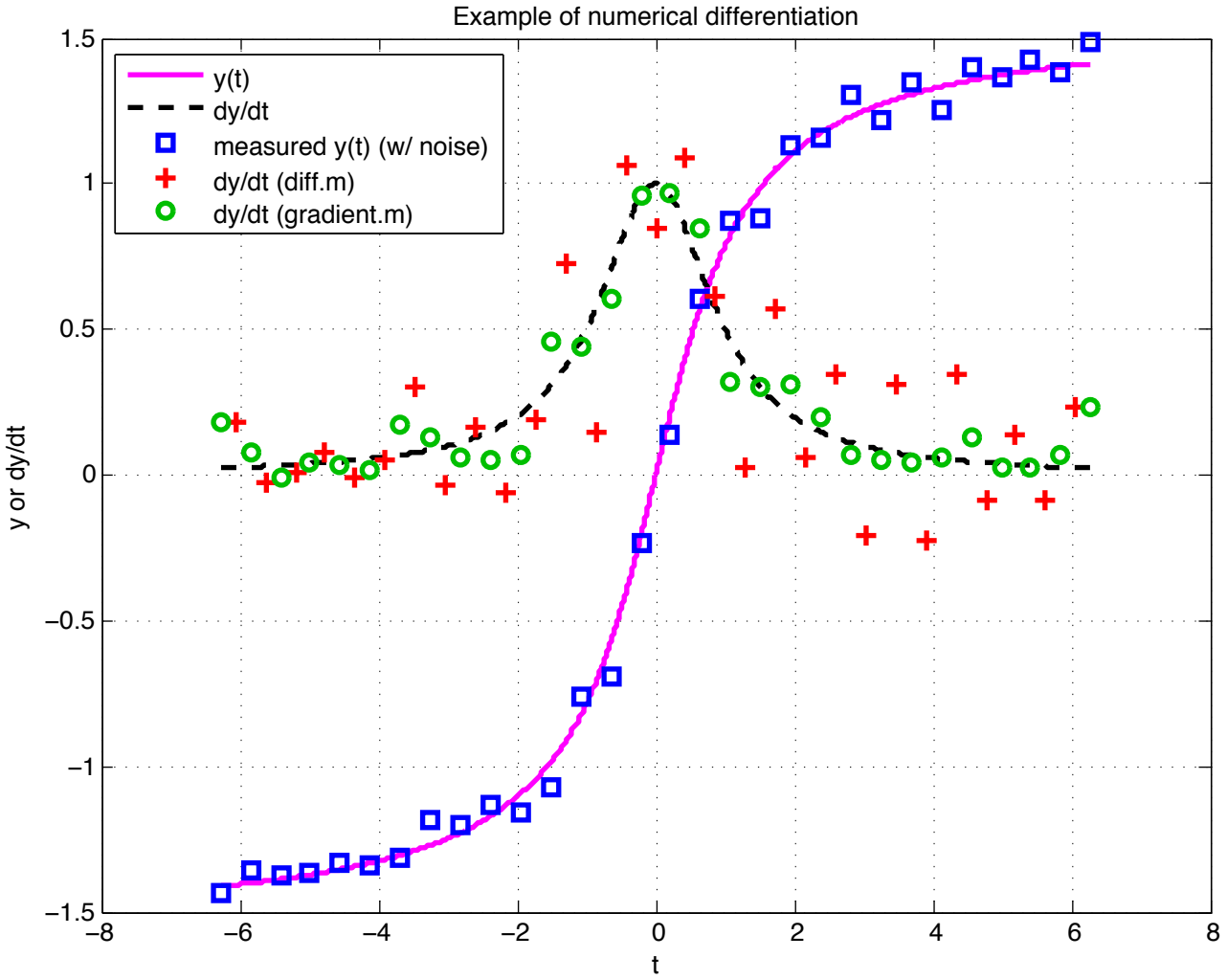
Note: When using cenetered differences, the quadratic error term subtracted out!

Figure 4.1: Graphical representation of the second-order accurate method for calculating the derivative with finite differences. The slope is simply rise over run where the nearest neighbors are used to determine both quantities. The specific function considered is $y = -\cos(t)$ with the derivative being calculated at $t = 3$ with $\Delta t = 0.5$.

Kutz (2013)

EXdiff1.m

Review: Numerical differentiation



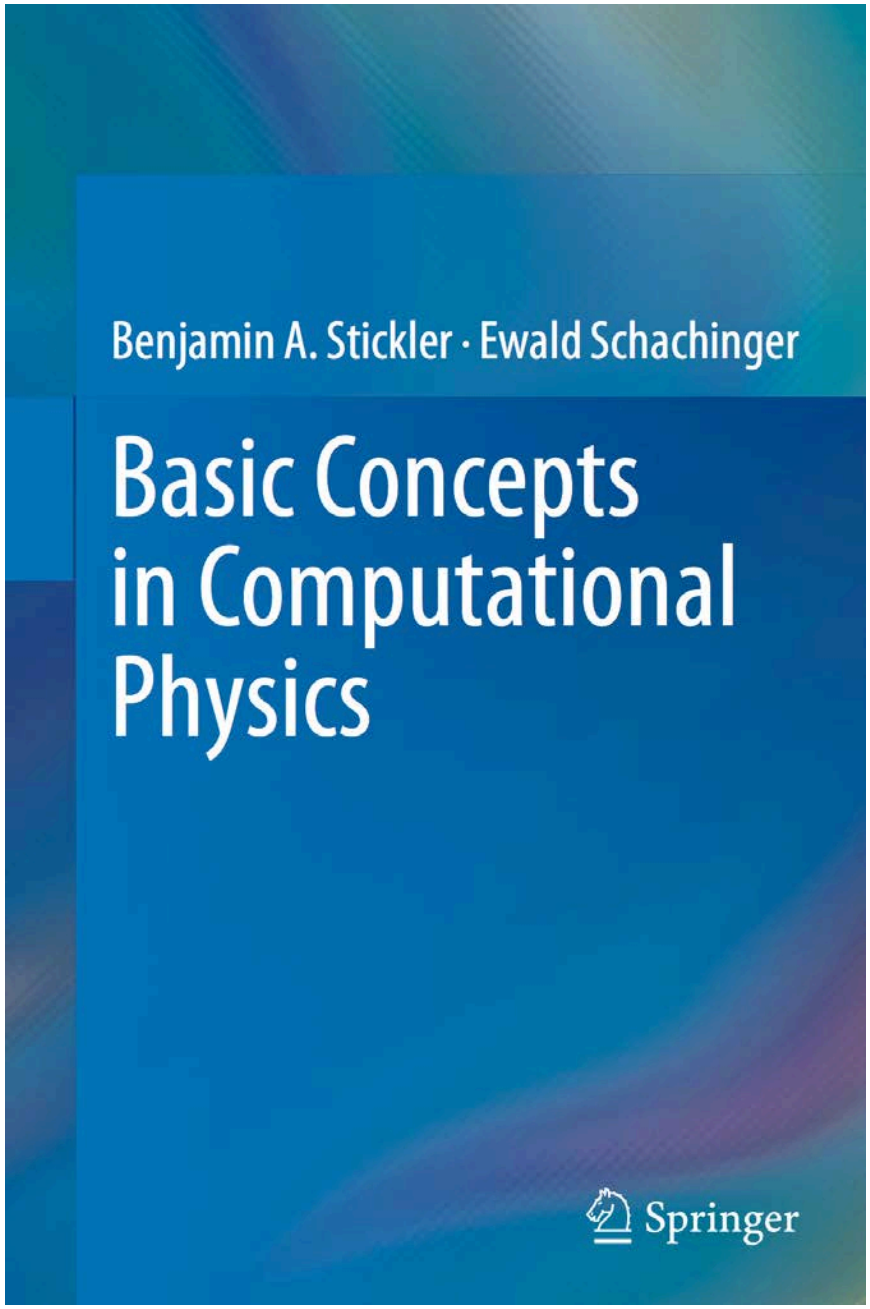

```
slope= gradient(y,t);
plot(t,slope,'go')
```

⇒ Improvement when using ‘centered differences’ (via gradient.m)

⇒ Be careful! Decreasing step size can ultimately lead to increased error

(‘rounding error’; see text)

French (1971)


Benjamin A. Stickler · Ewald Schachinger
Basic Concepts in Computational Physics
Springer

Chapter

Numerical Differentiation

Benjamin A. Stickler, Ewald Schachinger

» **Download PDF** (222KB) » **View Chapter** Pages 17-28

Chapter

Numerical Integration

Benjamin A. Stickler, Ewald Schachinger

» **Download PDF** (309KB) » **View Chapter** Pages 29-50

Chapter

The Kepler Problem

Benjamin A. Stickler, Ewald Schachinger

» **Download PDF** (186KB) » **View Chapter** Pages 51-59

Chapter

Ordinary Differential Equations: Initial Value Problems

Benjamin A. Stickler, Ewald Schachinger

» **Download PDF** (449KB) » **View Chapter** Pages 61-79

Chapter

The Double Pendulum

Benjamin A. Stickler, Ewald Schachinger

» **Download PDF** (2474KB) » **View Chapter** Pages 81-96

Chapter

Molecular Dynamics

Benjamin A. Stickler, Ewald Schachinger

» **Download PDF** (316KB) » **View Chapter** Pages 97-109

Chapter

Numerics of Ordinary Differential Equations: Boundary Value Problems

Benjamin A. Stickler, Ewald Schachinger

» **Download PDF** (176KB) » **View Chapter** Pages 111-122

Chapter

The One-Dimensional Stationary Heat Equation

Benjamin A. Stickler, Ewald Schachinger

» **Download PDF** (297KB) » **View Chapter** Pages 123-129

Chapter

The One-Dimensional Stationary SCHRÖDINGER Equation

Benjamin A. Stickler, Ewald Schachinger

» **Download PDF** (445KB) » **View Chapter** Pages 131-146

Chapter

Partial Differential Equations

Benjamin A. Stickler, Ewald Schachinger

» **Download PDF** (831KB) » **View Chapter** Pages 147-168

Stochastic Methods

Front Matter

» **Download PDF** (34KB) Pages 169-169

Chapter

Pseudo Random Number Generators

Benjamin A. Stickler, Ewald Schachinger

» **Download PDF** (364KB) » **View Chapter** Pages 171-183

Numerical integration

Numerical integration simply calculates the area under a given curve. The basic ideas for performing such an operation come from the definition of integration

$$\int_a^b f(x)dx = \lim_{h\to 0} \sum_{j=0}^{N} f(x_j)h \tag{4.2.1}$$

where $b - a = Nh$. Thus the area under the curve, from the calculus standpoint, is thought of as a limiting process of summing up an ever-increasing number of rectangles. This process is known as numerical quadrature.

Kutz (2013)

Key idea(s)

$$\int_a^b f(x)dx = \lim_{h\to 0} \sum_{j=0}^{N} f(x_j)h \qquad (4.2.1)$$

- Distinction between 'continuous' vs 'discrete'
  (or somewhat similarly, 'analog' versus 'digital')

- Computationally, we cannot truly achieve such a limit and more realistically have:

$$\int_a^b f(x)dx \approx \sum_{j=0}^{N} f(x_j)h$$

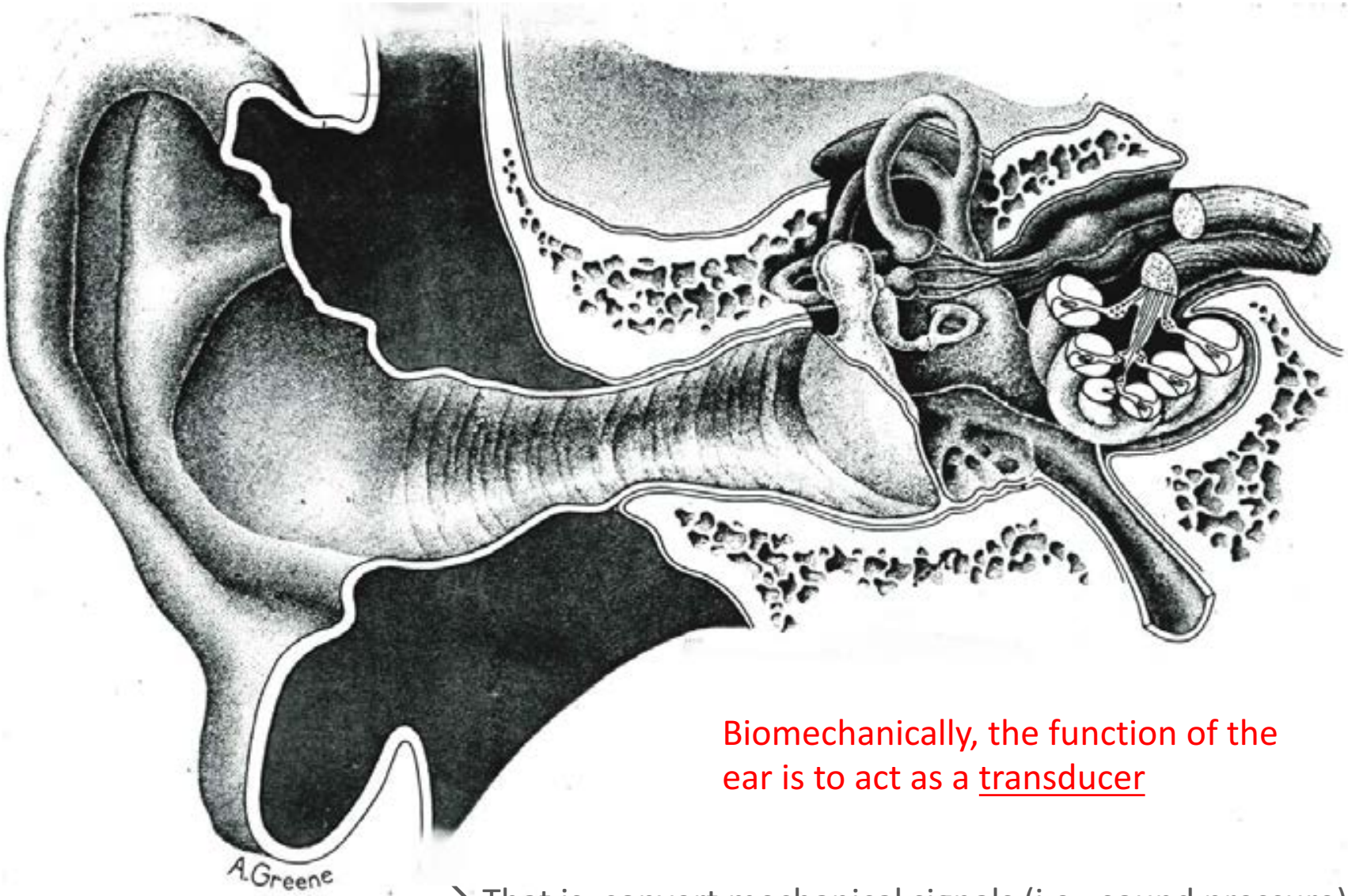
for very small h (or, large N spanning over The range $x=[a,b]$)

- Brute force, chopping up into very small segments can improve our approximation. Conversely, 'higher order' methods can further help.

Our approach:

i. Motivate a real world problem (related to hearing and neurophysiology)
ii. Review the notion of Riemann sums
iii. Look at example of computational framework

Kutz (2013)



Biomechanically, the function of the ear is to act as a transducer

→ That is, convert mechanical signals (i.e., sound pressure) to electrical ones (i.e., action potentials going in towards brainstem)

Tuning

http://houndingsqueegee.files.wordpress.com/2009/02/piano-keys.jpg

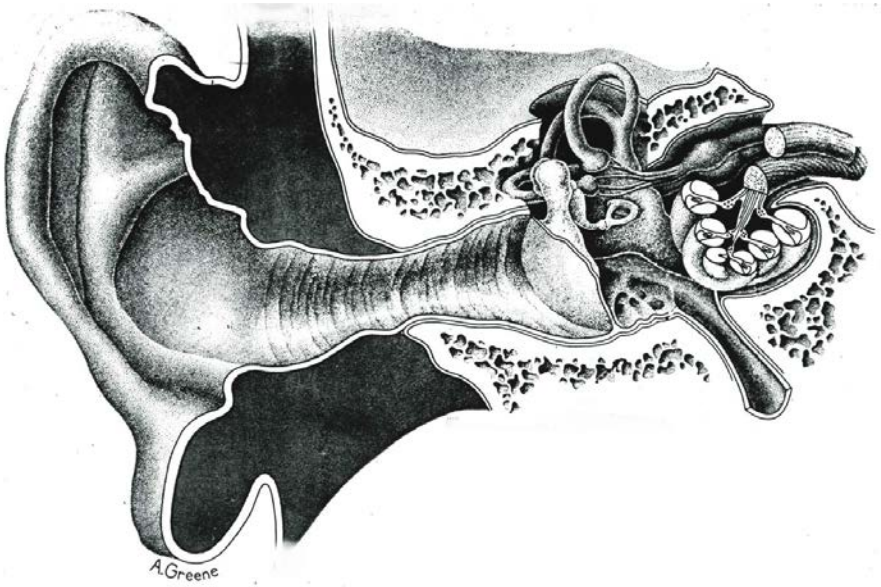
http://www.axebay.com/blog/love-your-strings

→ Ear is both *sensitive* & *selective*

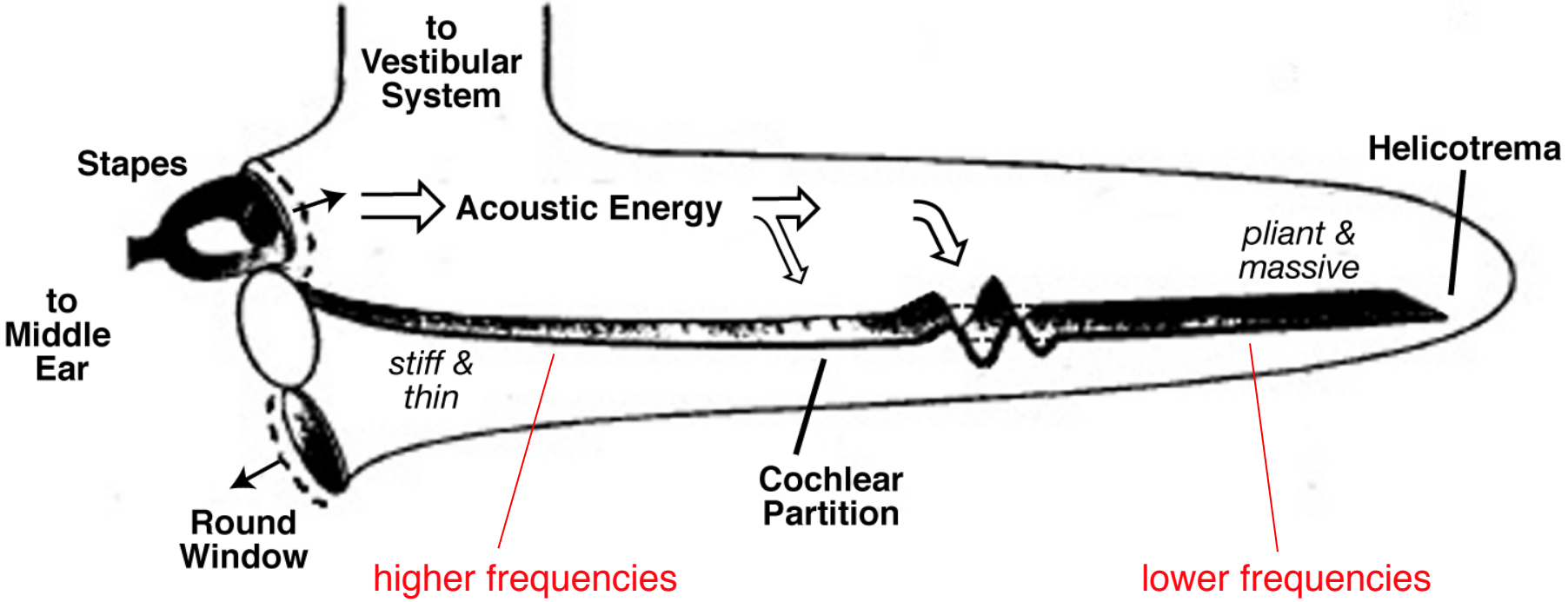
[e.g., think of a spectrum analyzer or an equalizer on your music device]

http://www.flickr.com/photos/athena/369037746/

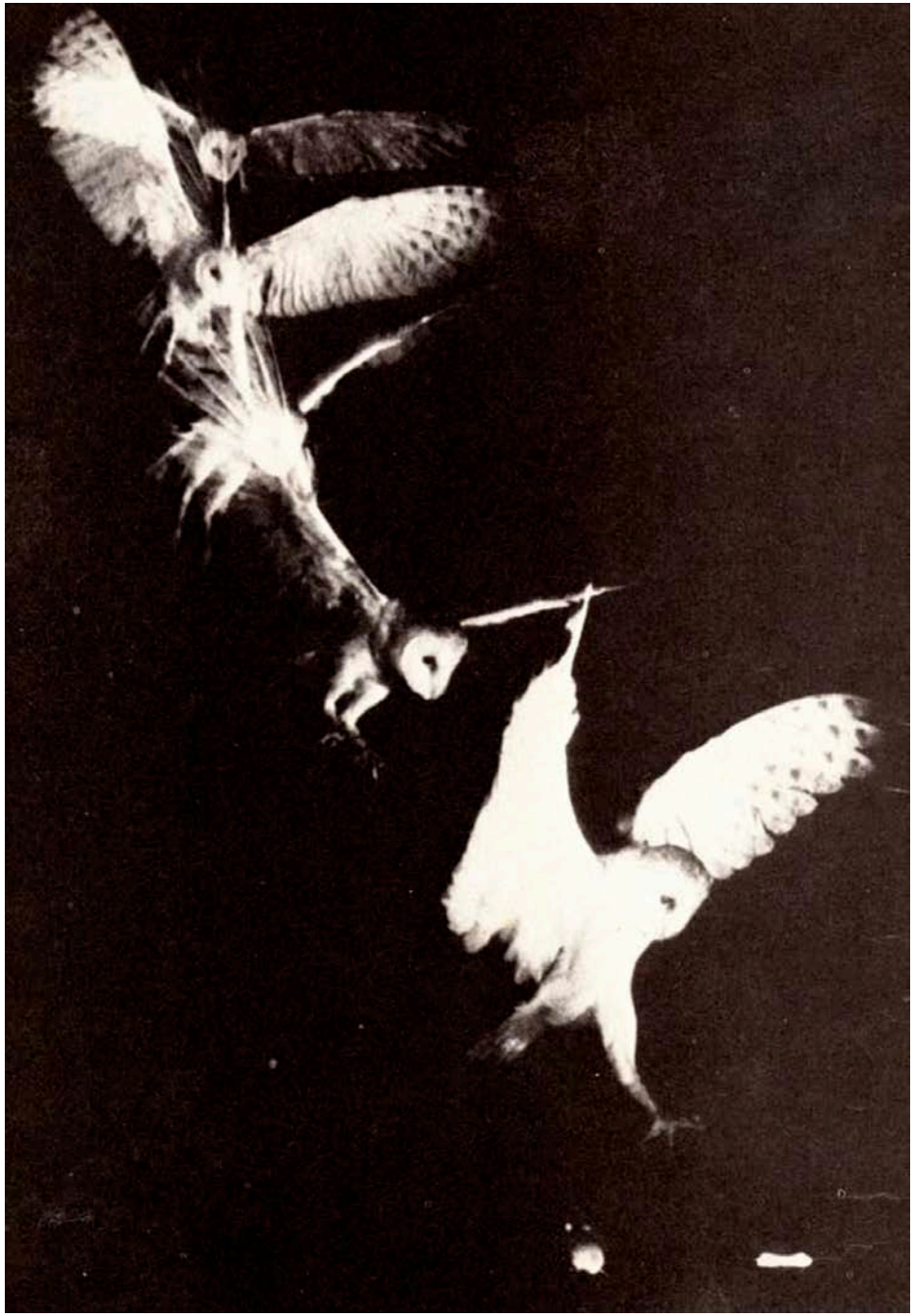
Ear is a Fourier analyzer (*Tonotopicity*)



Basilar membrane (BM)



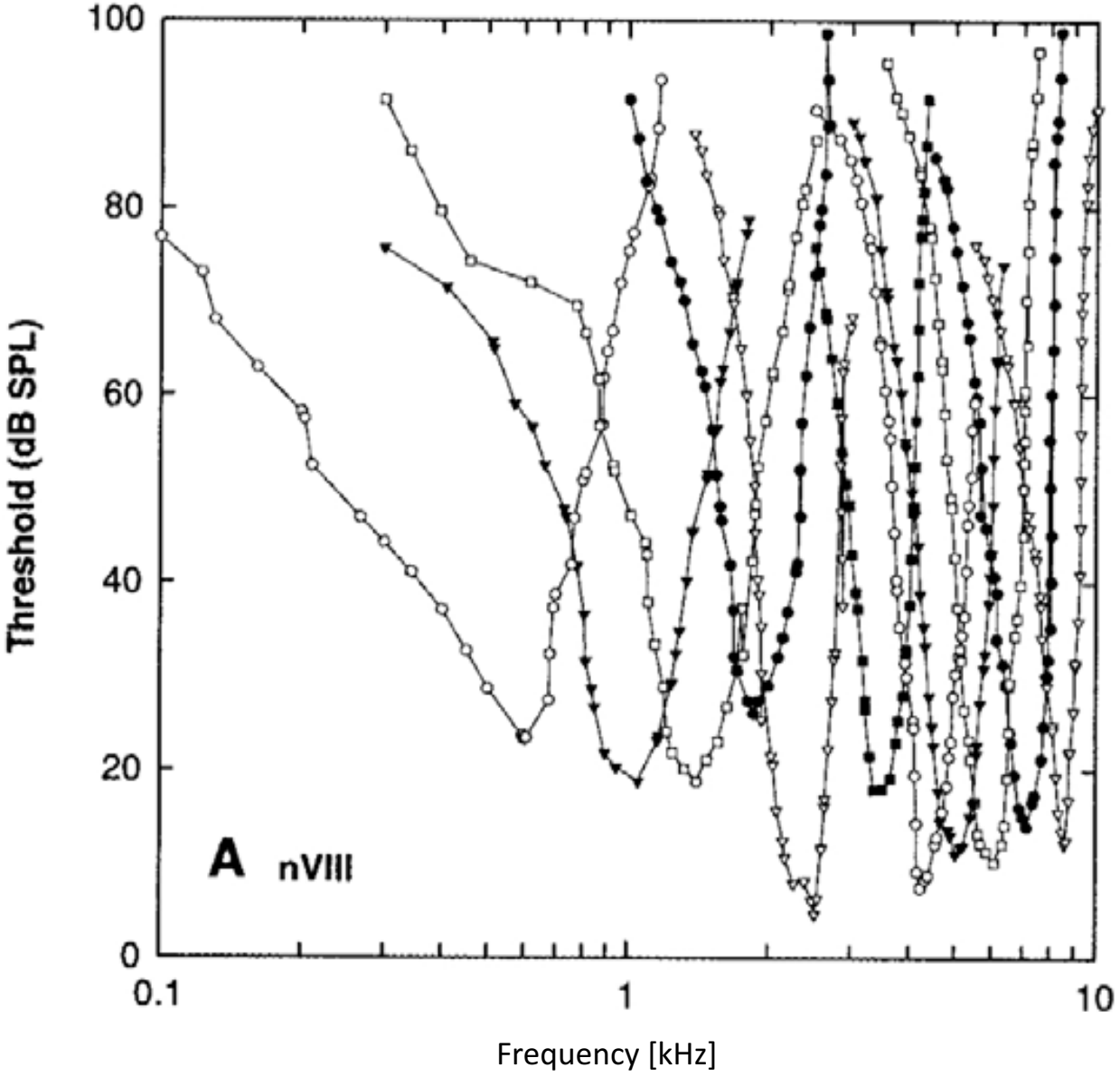
Geisler (1996)

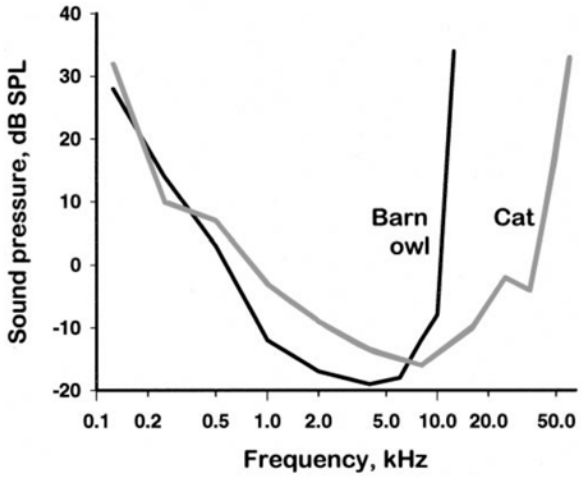


Konishi (1973)

Barn owl auditory neurophysiology

Auditory nerve fiber (single unit) threshold tuning curves

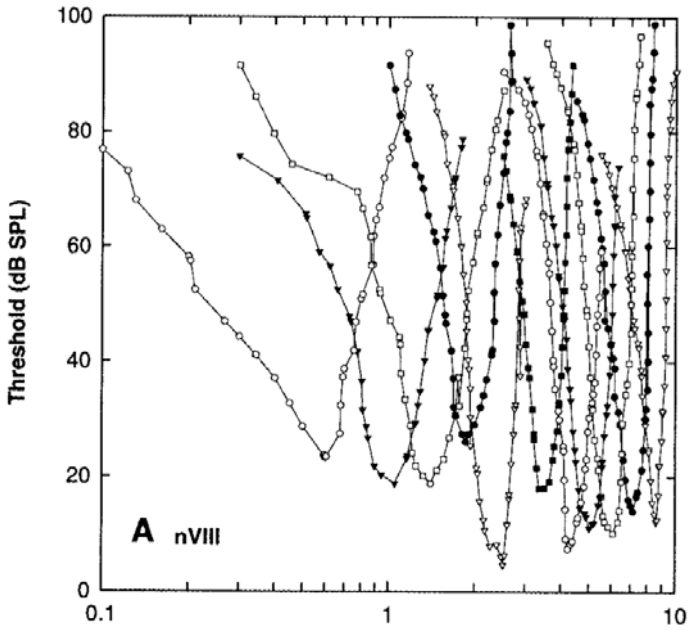




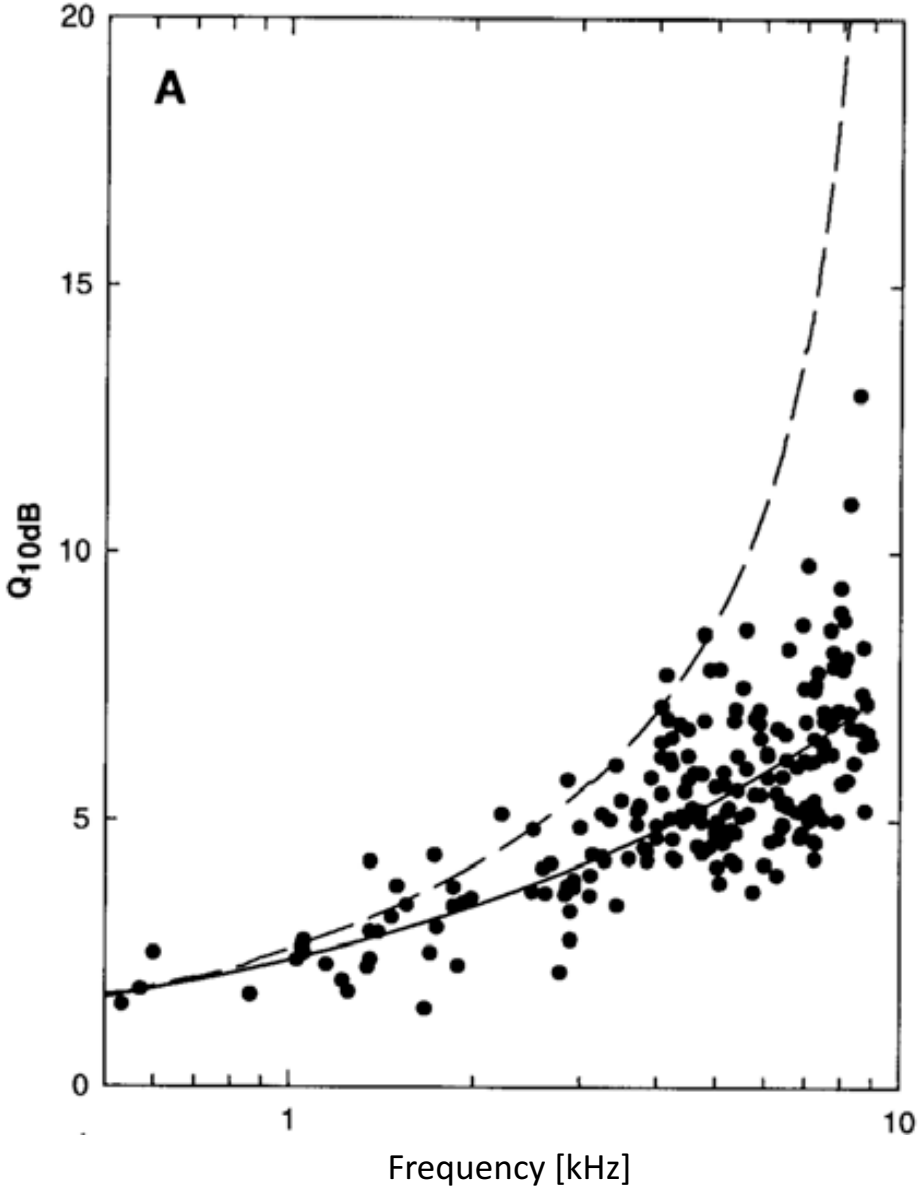
→ Match well to behavioral measures

Koppl (1997a)

Barn owl papilla neurophysiology



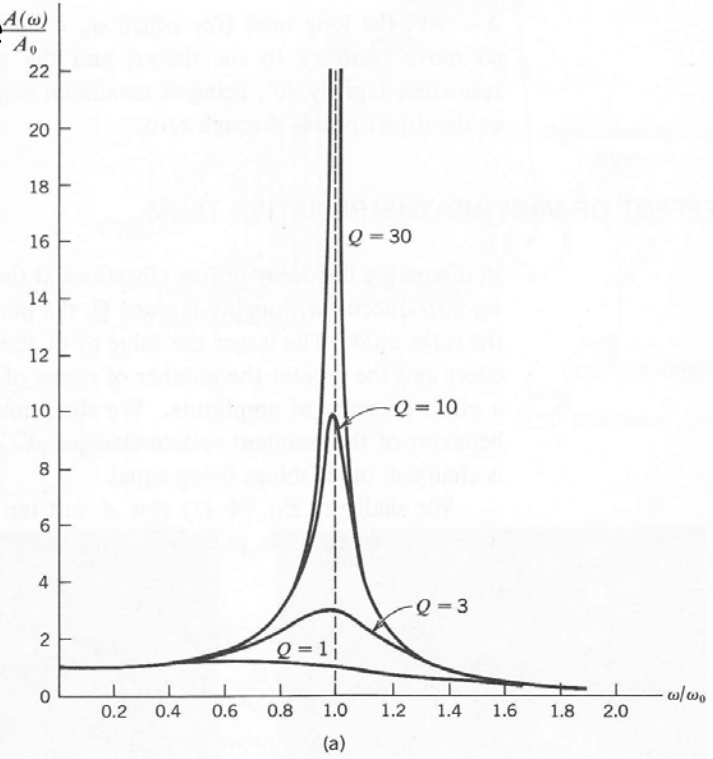
ANF 'quality factor' (Q)



Koppl (1997a)

Review: Resonance

Steady-state frequency response



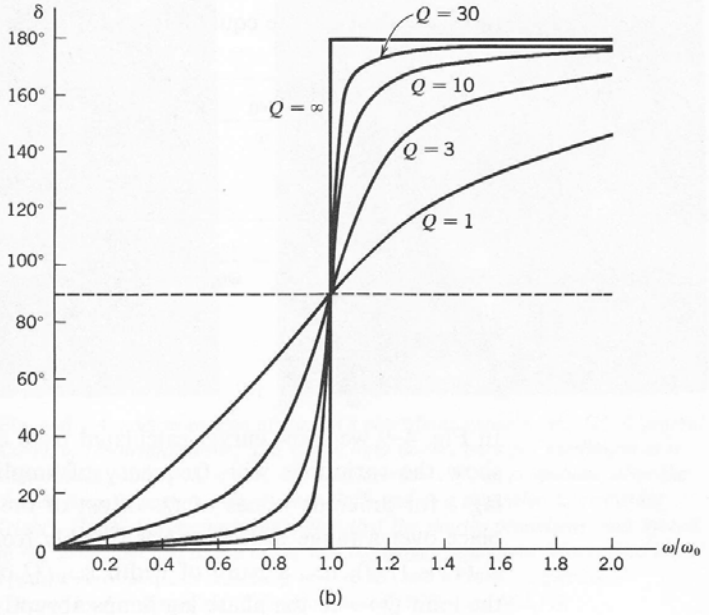


Fig. 4–9 (a) Amplitude as function of driving frequency for different values of Q, assuming driving force of constant magnitude but variable frequency. (b) Phase difference δ as function of driving frequency for different values of Q.

Consider the sinusoidally "driven" case:

$$\ddot{x} + \gamma\dot{x} + \omega_o^2 x = \frac{F_o}{m} e^{i\omega t}$$

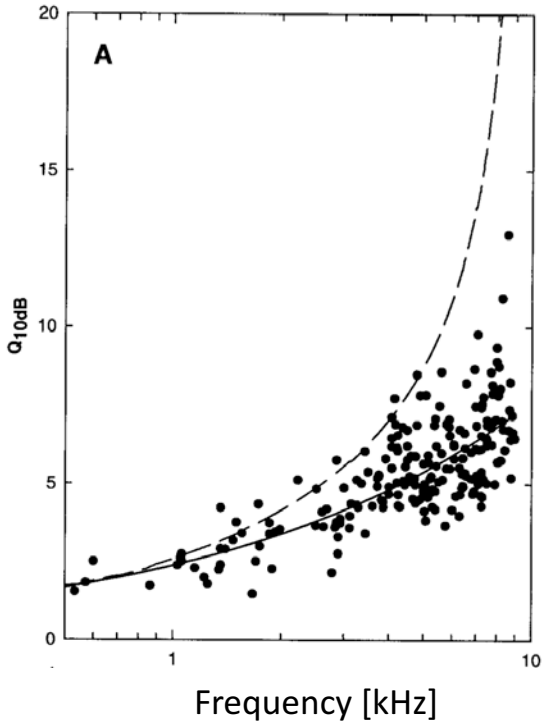
$Q = \omega_o / \gamma$ Q is the 'quality factor'

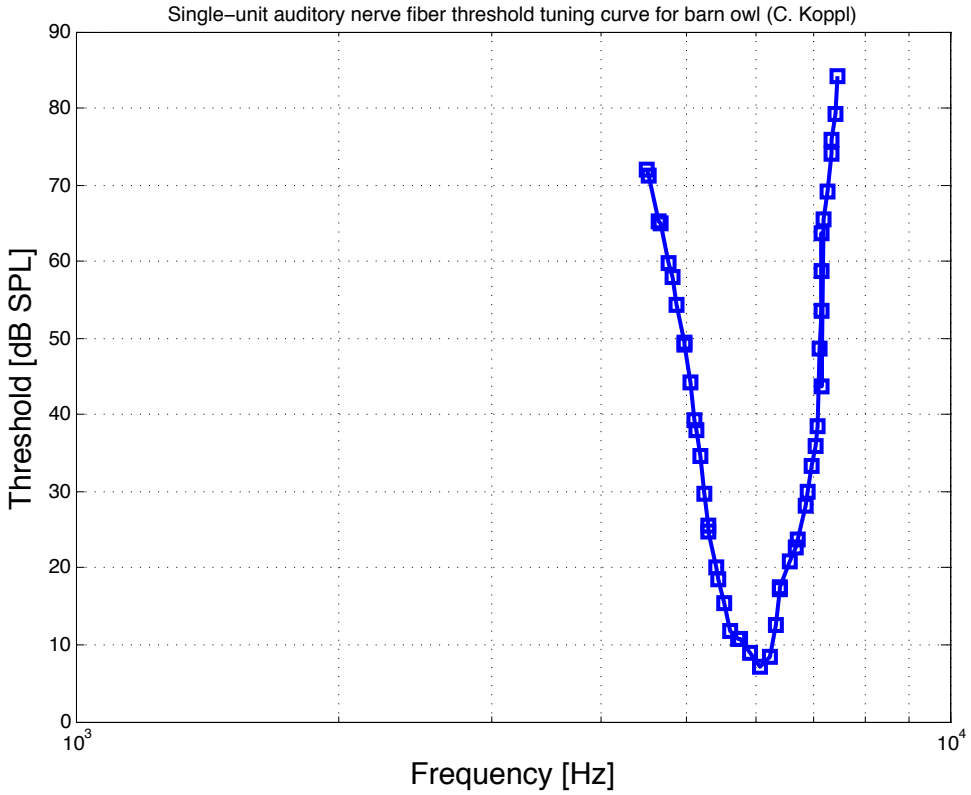
- The "system" stores energy, and more of it when driven near the natural frequency
- Less damping (i.e., higher Q) means more energy stored

French (1971)

Motivation: Numerical integration

ANF ‘quality factor’ (Q)





Question: How do we determine a Q-value from an individual tuning curves?

→ Numerical integration!

For additional background, examine the notion of a ‘critical band’ pioneered by Harvey Fletcher (http://en.wikipedia.org/wiki/Critical_band)

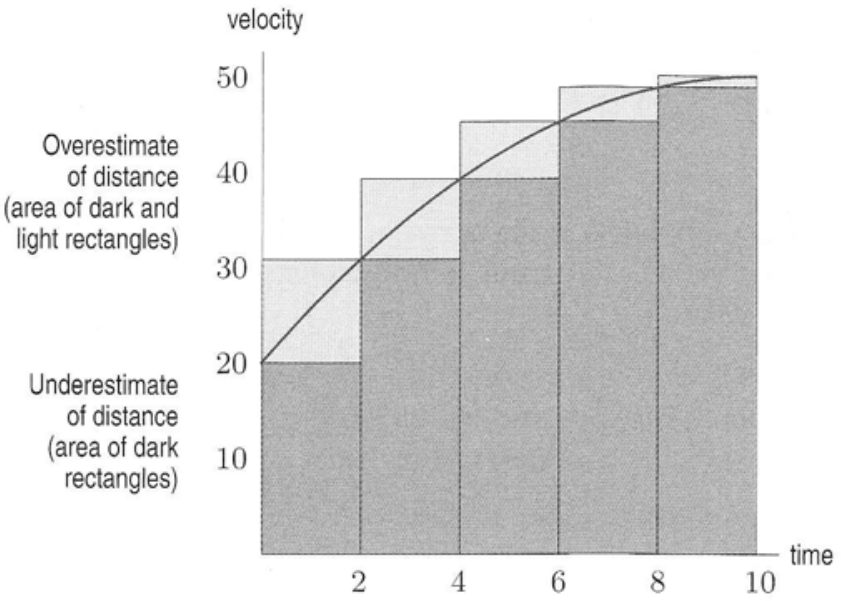
Background: Riemann sums

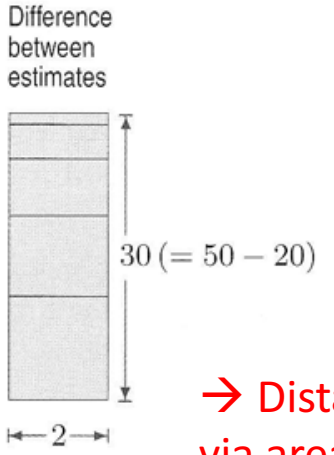
ex. distance covered by a moving car

- At constant velocity, then distance = velocity x time
- But if velocity is changing with time.....

Suppose velocity is measured every 2 seconds

Time (sec)	0	2	4	6	8	10
Velocity (ft/sec)	20	30	38	44	48	50





→ Distance can be approximated via area of triangle, indicating a bound of 360-420 ft

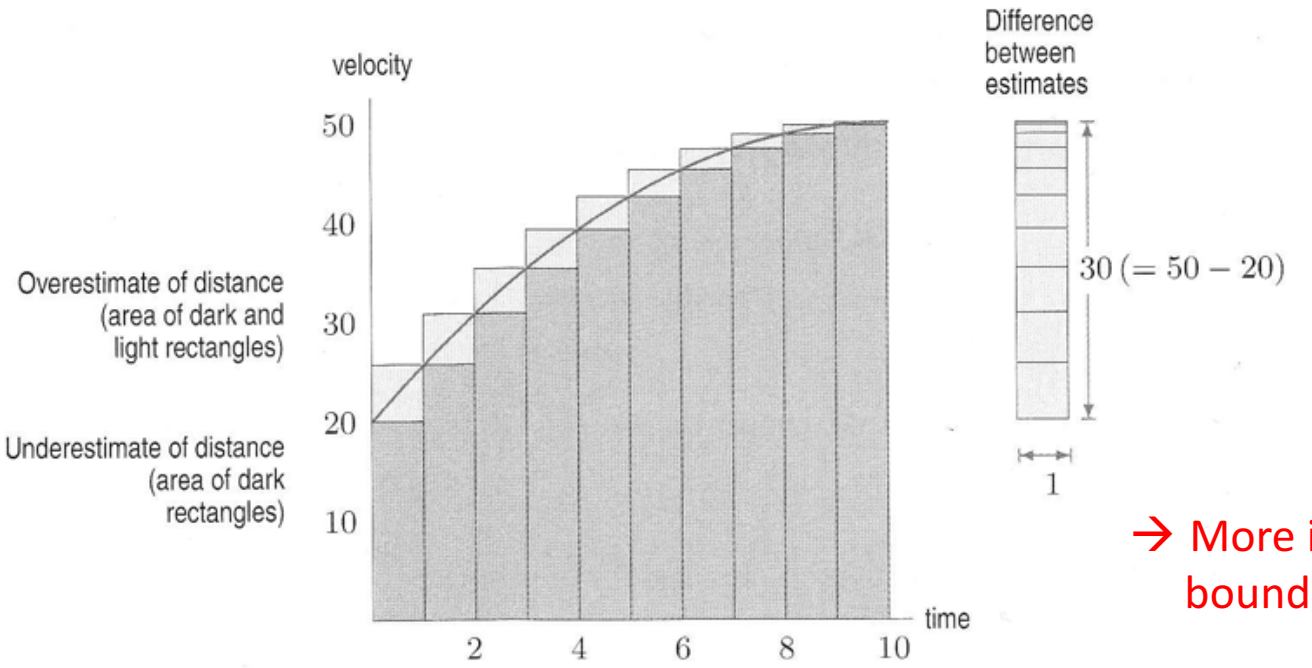
Hughes-Hallett et al. (2005)

Background: Riemann sums

ex. distance covered by a moving car

Time (sec)	0	1	2	3	4	5	6	7	8	9	10
Velocity (ft/sec)	20	26	30	34	38	41	44	46	48	49	50

Suppose velocity is measured every 1 seconds



→ More info leads to tighter bounds: 376-406 ft

Hughes-Hallett et al. (2005)

Background: Riemann sums

→ More info leads to tighter bounds

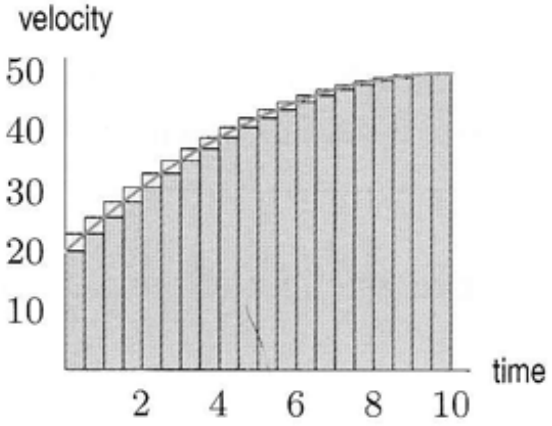


Figure 5.3: Velocity measured every 1/2 second

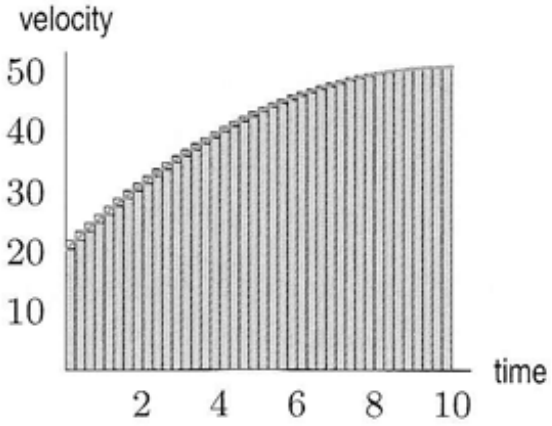


Figure 5.4: Velocity measured every 1/4 second

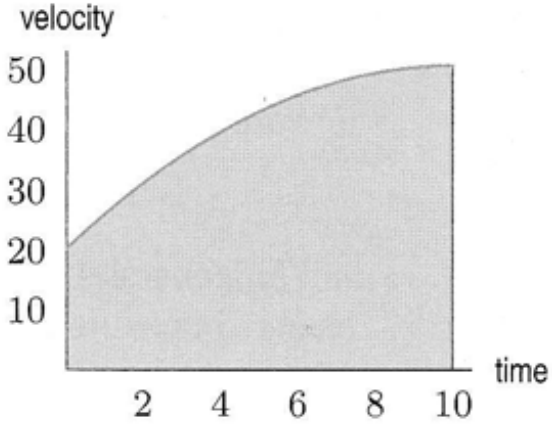


Figure 5.5: Distance traveled is area under curve

Note: Velocity could be negative as well!

Hughes-Hallett et al. (2005)

Background: Riemann sums

Assumption: $f(t)$ is continuously differentiable over interval $[a,b]$

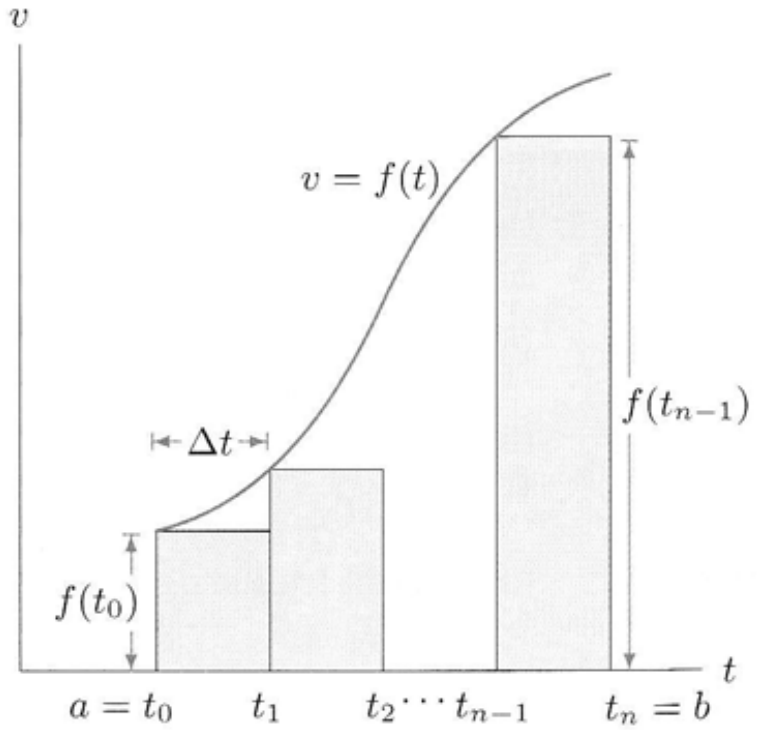


Figure 5.8: Left-hand sums

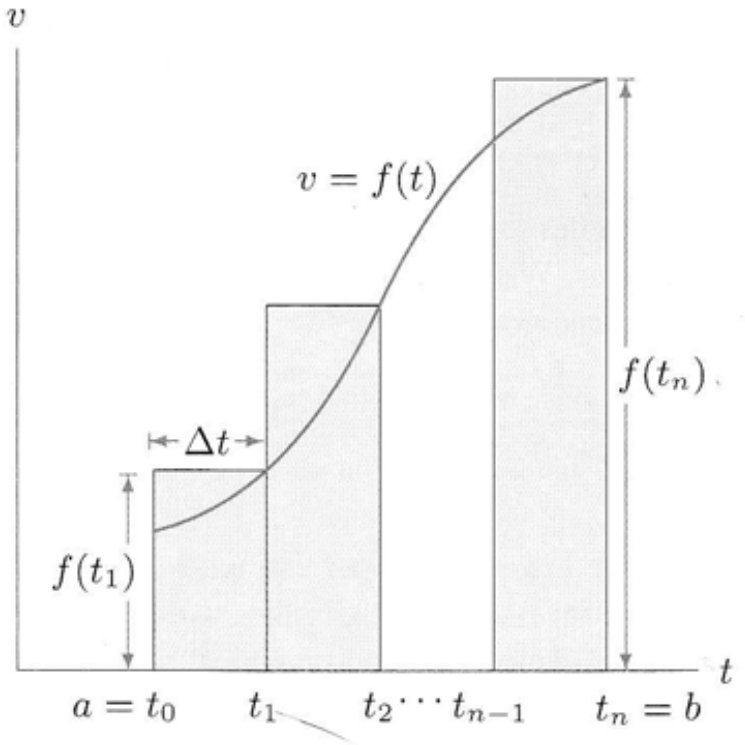


Figure 5.9: Right-hand sums

Time step

$$\Delta t = \frac{b-a}{n}.$$

$$\text{Left-hand sum} = f(t_0)\Delta t + f(t_1)\Delta t + \cdots + f(t_{n-1})\Delta t = \sum_{i=0}^{n-1} f(t_i)\Delta t.$$

$$\text{Right-hand sum} = f(t_1)\Delta t + f(t_2)\Delta t + \cdots + f(t_n)\Delta t = \sum_{i=1}^{n} f(t_i)\Delta t.$$

Hughes-Hallett et al. (2005)

Background: Riemann sums

Suppose f is continuous for $a \le t \le b$. The **definite integral** of f from a to b, written

$$\int_a^b f(t)\,dt,$$

is the limit of the left-hand or right-hand sums with n subdivisions of $a \le t \le b$ as n gets arbitrarily large. In other words,

$$\int_a^b f(t)\,dt = \lim_{n\to\infty} (\text{Left-hand sum}) = \lim_{n\to\infty} \left(\sum_{i=0}^{n-1} f(t_i)\Delta t \right)$$

and

$$\int_a^b f(t)\,dt = \lim_{n\to\infty} (\text{Right-hand sum}) = \lim_{n\to\infty} \left(\sum_{i=1}^{n} f(t_i)\Delta t \right).$$

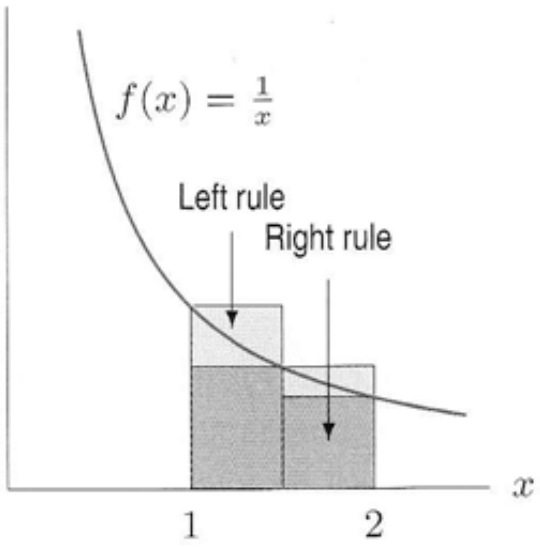
Each of these sums is called a *Riemann sum*, f is called the *integrand*, and a and b are called the *limits of integration*.

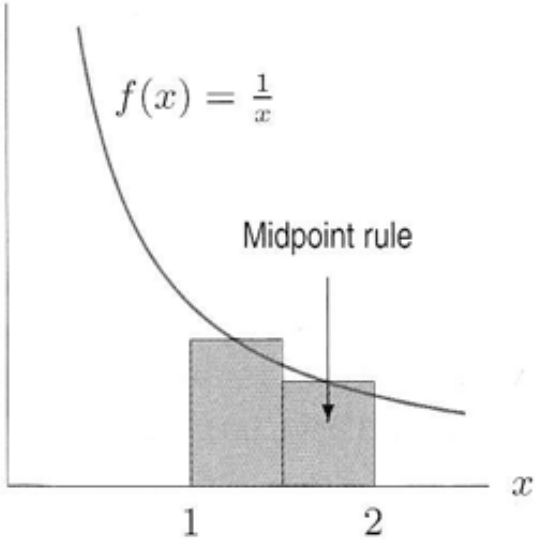
The "$\int$" notation comes from an old-fashioned "S," which stands for "sum" in the same way that $\sum$ does. The "dt" in the integral comes from the factor Δt. Notice that the limits on the $\sum$ symbol are 0 and $n-1$ for the left-hand sum, and 1 and n for the right-hand sum, whereas the limits on the $\int$ sign are a and b.

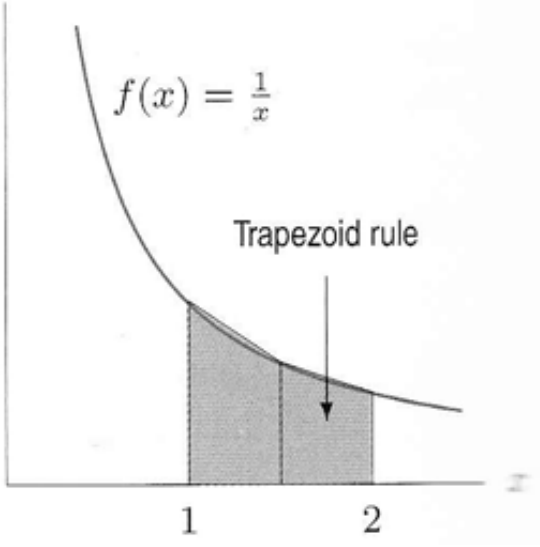
Hughes-Hallett et al. (2005)

Background: Riemann sums

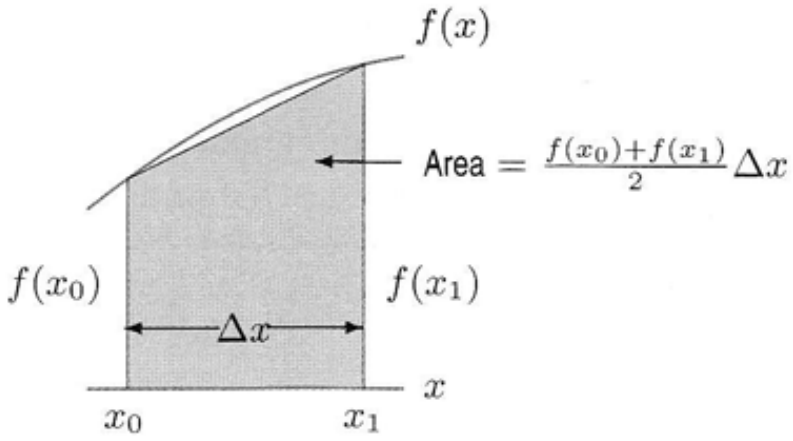
Other methods







$$\mathrm{TRAP}(n) = \frac{\mathrm{LEFT}(n) + \mathrm{RIGHT}(n)}{2}$$



Hughes-Hallett et al. (2005)

Background: Errors in riemann sums

Another example

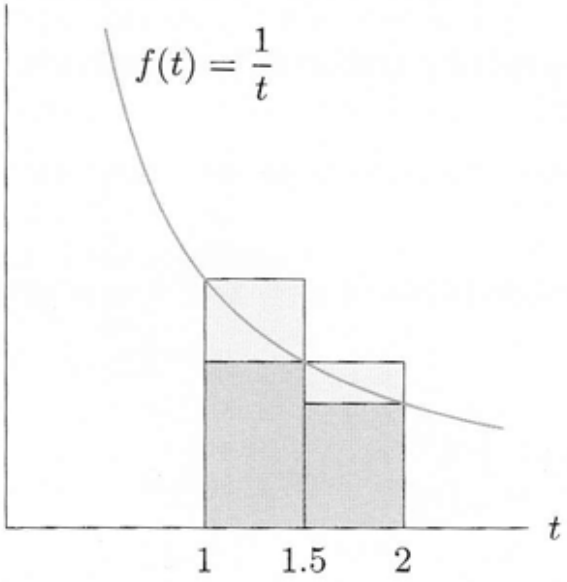


Figure 5.15: Approximating $\int_1^2 \frac{1}{t}\,dt$ with $n = 2$

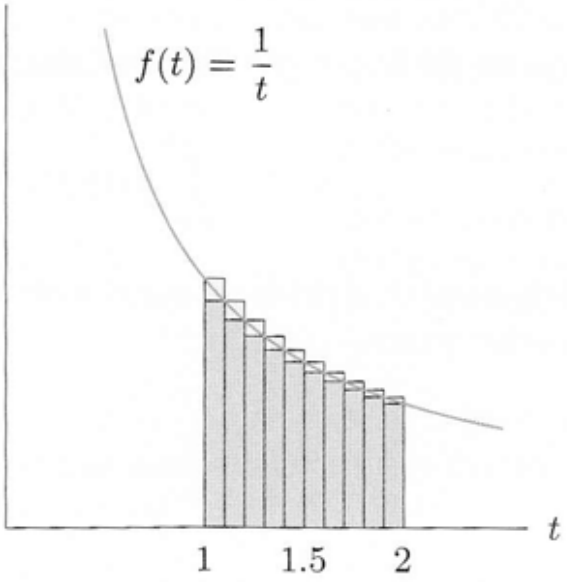


Figure 5.16: Approximating $\int_1^2 \frac{1}{t}\,dt$ with $n = 10$

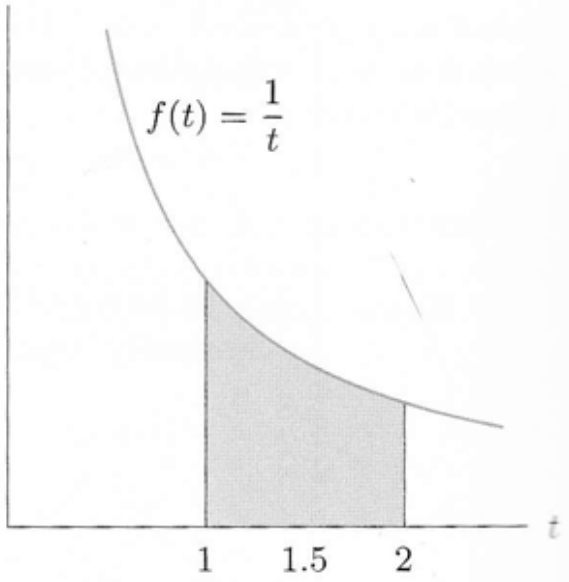
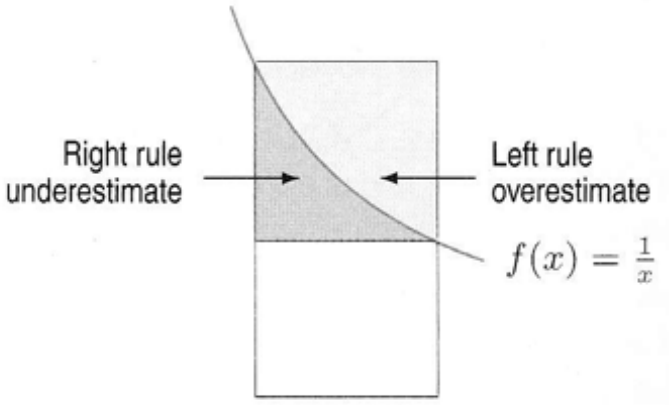


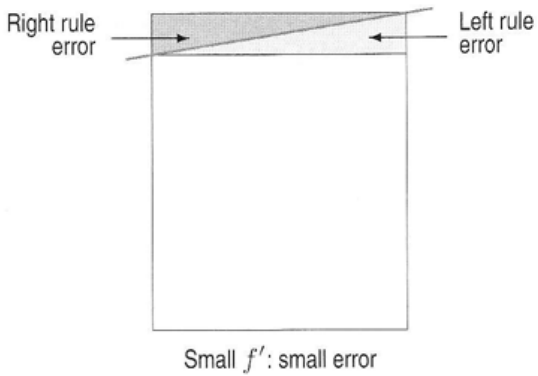
Figure 5.17: Shaded area is exact value of $\int_1^2 \frac{1}{t}\,dt$

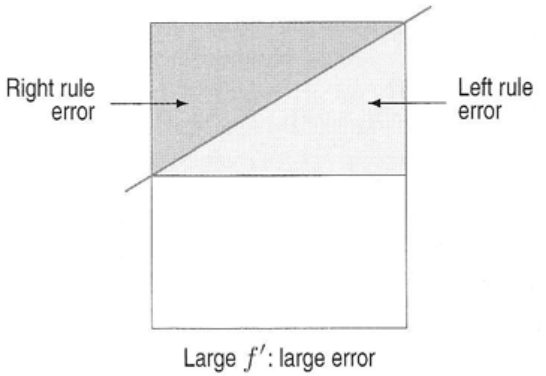
→ Note that decreasing Δt appears to decrease the overall 'error'

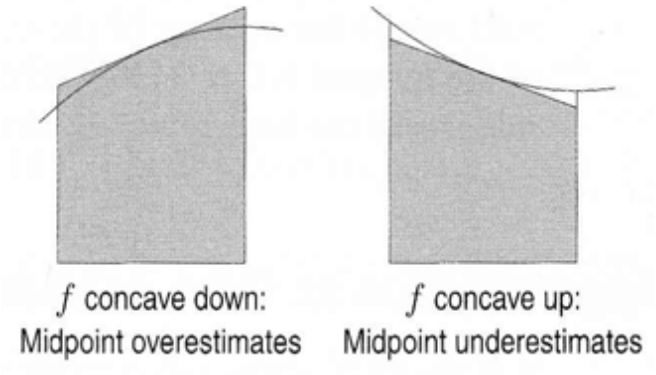
Hughes-Hallett et al. (2005)

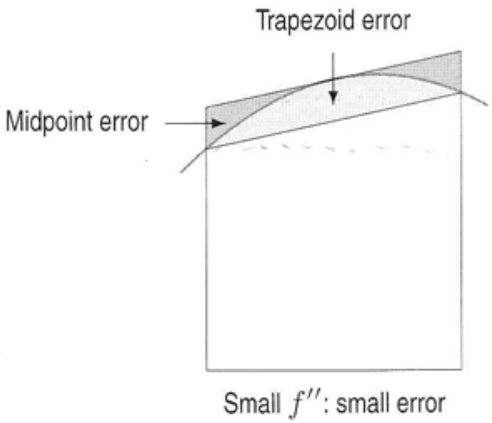
Background: Errors in riemann sums

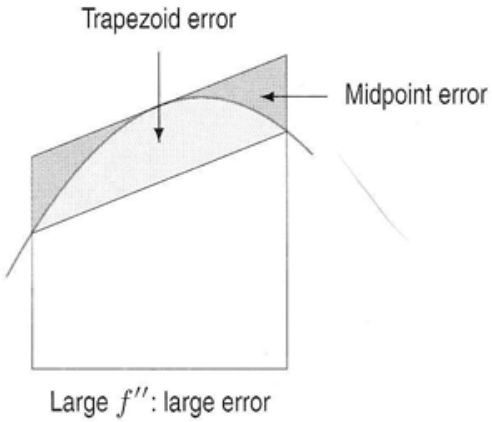






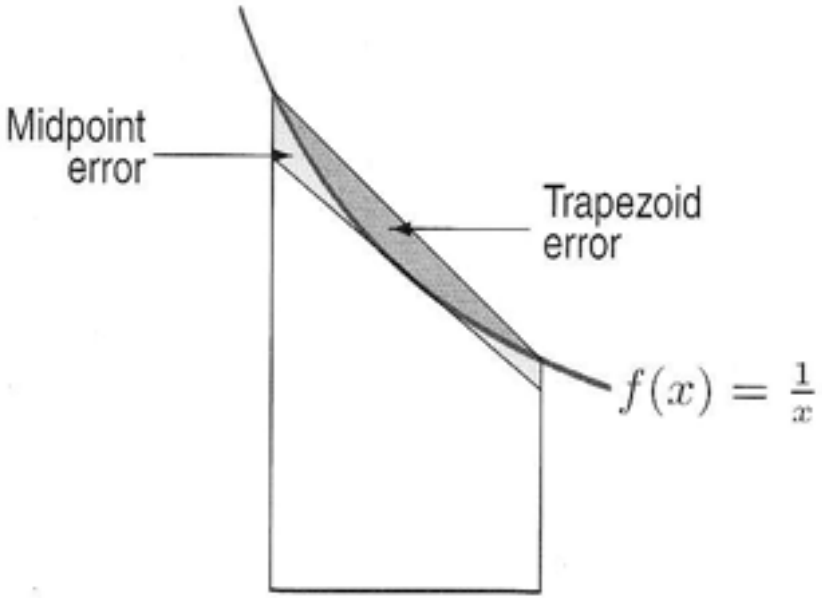






Hughes-Hallett et al. (2005)

Can we get errors to cancel one another out?



Simpson's rule

$$\mathrm{SIMP}(n) = \frac{2 \cdot \mathrm{MID}(n) + \mathrm{TRAP}(n)}{3}$$

Hughes-Hallett et al. (2005)

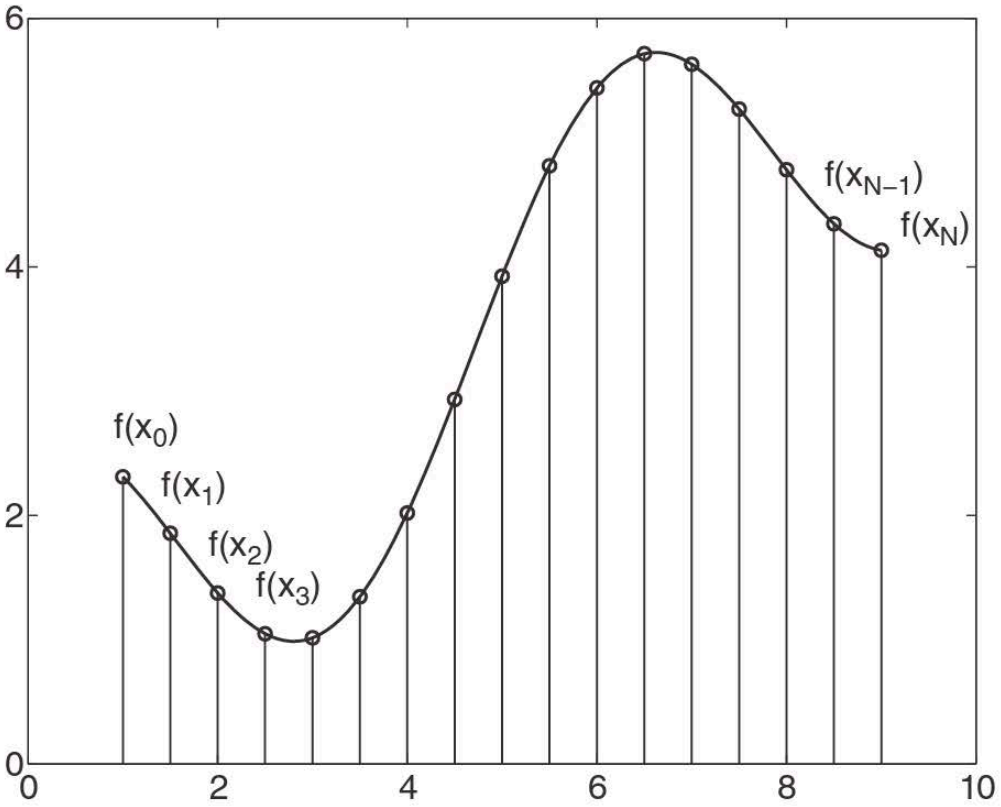


Figure 4.3: Graphical representation of the integration process. The integration interval is broken up into a finite set of points. A quadrature rule then determines how to sum up the area of a finite number of rectangles.

Kutz (2013)

Newton-Cotes Formulae

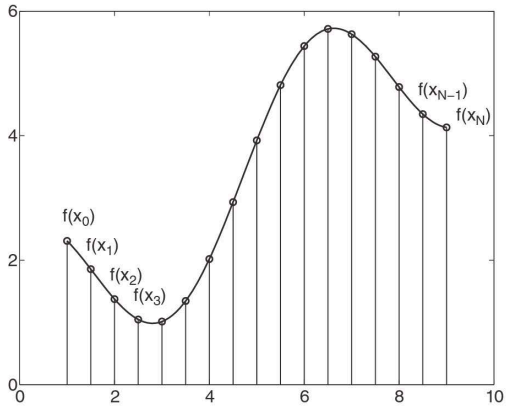


Figure 4.3: Graphical representation of the integration process. The integration interval is broken up into a finite set of points. A quadrature rule then determines how to sum up the area of a finite number of rectangles.

$$\text{Trapezoid rule } \int_{x_0}^{x_1} f(x)dx = \frac{h}{2}(f_0 + f_1) - \frac{h^3}{12}f''(c) \qquad (4.2.6a)$$

$$\text{Simpson's rule } \int_{x_0}^{x_2} f(x)dx = \frac{h}{3}(f_0 + 4f_1 + f_2) - \frac{h^5}{90}f''''(c) \qquad (4.2.6b)$$

$$\text{Simpson's 3/8 rule } \int_{x_0}^{x_3} f(x)dx = \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3) - \frac{3h^5}{80}f''''(c) \qquad (4.2.6c)$$

$$\text{Boole's rule } \int_{x_0}^{x_4} f(x)dx = \frac{2h}{45}(7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4) - \frac{8h^7}{945}f^{(6)}(c). \qquad (4.2.6d)$$

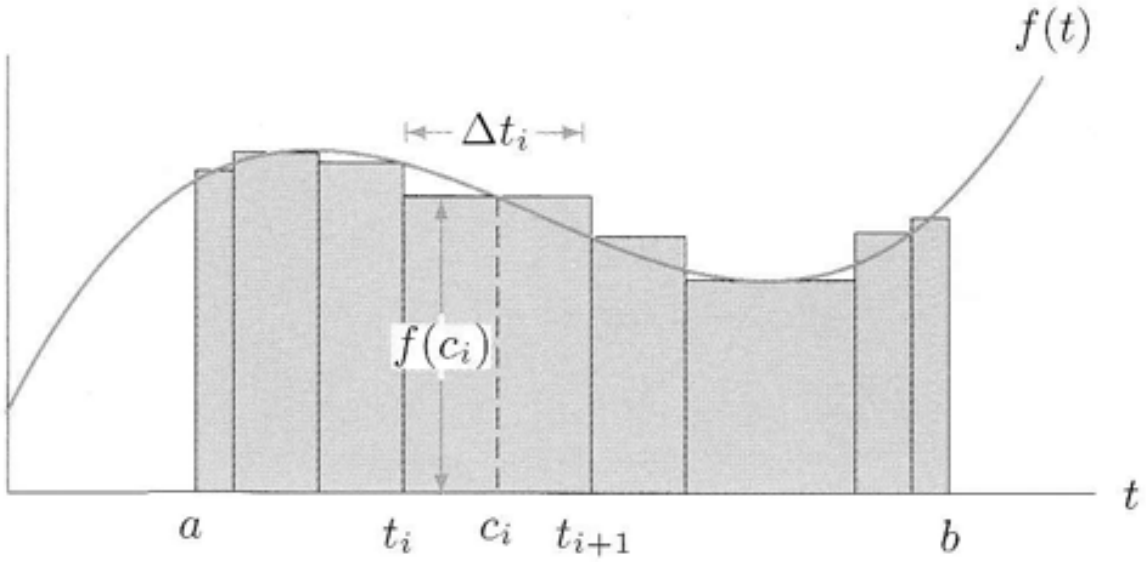
Note that these formulae have an extra 'error' term

$$P_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 \qquad (4.2.4)$$

→ Ultimately, we are approximating via a series of polynomials
(and we decide how high we want to go!)

Kutz (2013)

Background: Riemann sums



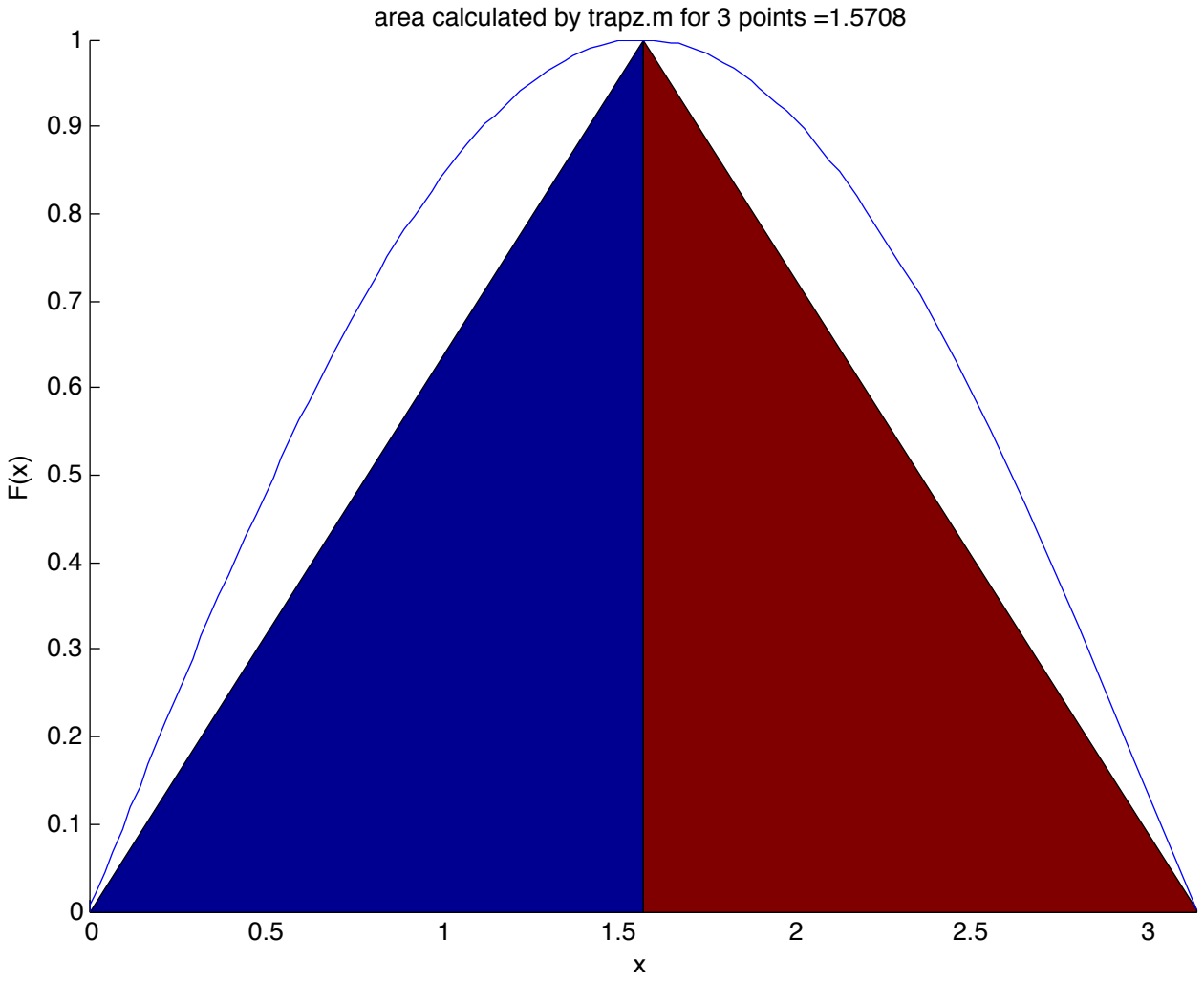
→ Note that Δt need not be constant (this is helpful computationally!)

Hughes-Hallett et al. (2005)

```
% Numerical integration example - original source:
% http://ef.engr.utk.edu/ef230-2011-01/modules/matlab-integration/

clear;
% ----------------------
% User parameters
F = @(x)(sin(x)); % function to integrate
%F = @(x)(exp(-x.^2/2)); % function to integrate
xL= [0 pi]; % integration limits

N= 5;    % Method A - # of points for LEFT and RIGHT
pts= [3 4 5 10 25]; % Method B - # of points to consider integrating (via trapz function)
dur= 1;     % Method B - pause duration [s] for trapz loop
% ----------------------

% ***************
% Show the curve
figure(1);
fplot(F,[xL(1),xL(2)]) % a quick way to plot a function
xlabel('x'); ylabel('F(x)');

% ***************
% Method A
% Approximate the integral via brute force LEFT and RIGHT Riemann sums
sumL= 0; sumR=0;
delX= (xL(2)-xL(1))/N;    % step-size
x= linspace(xL(1),xL(2),N+1);  % add one since N is # of 'boxes' and is really N-1
for nn=1:N
    sumL= sumL + F(x(nn))*delX;
    sumR= sumR + F(x(nn+1))*delX;
end
disp(['left-hand rule yields =',num2str(sumL),' (for ',num2str(N),' steps)']);
disp(sprintf('right-hand rule yields = %g', sumR));

% ***************
% Method B
% Approximate the integral via trapz for different numbers of points
for np=pts
    figure(2); clf % clear the current figure
    hold on % allow stuff to be added to this plot
    x = linspace(xL(1),xL(2),np); % generate x values
    y = F(x); % generate y values
    a2 = trapz(x,y); % use trapz to integrate
    % Generate and display the trapezoids used by trapz
    for ii=1:length(x)-1
        px=[x(ii) x(ii+1) x(ii+1) x(ii)];   py=[0 0 y(ii+1) y(ii)];
        fill(px,py,ii)
    end
    fplot(F,[xL(1),xL(2)]); xlabel('x'); ylabel('F(x)');
    disp(['area calculated by trapz.m for ',num2str(np),' points =',num2str(a2)]);
    title(['area calculated by trapz.m for ',num2str(np),' points =',num2str(a2)]);
    pause(dur);   % wait a bit
end

% ***************
% Method C
a1 = quad(F,xL(1),xL(2)); % use quad to integrate
msg = ['area calculated by quad.m = ' num2str(a1,10)]; disp(msg);
```

What three different methods are being used? Which ones are a ‘black box’?

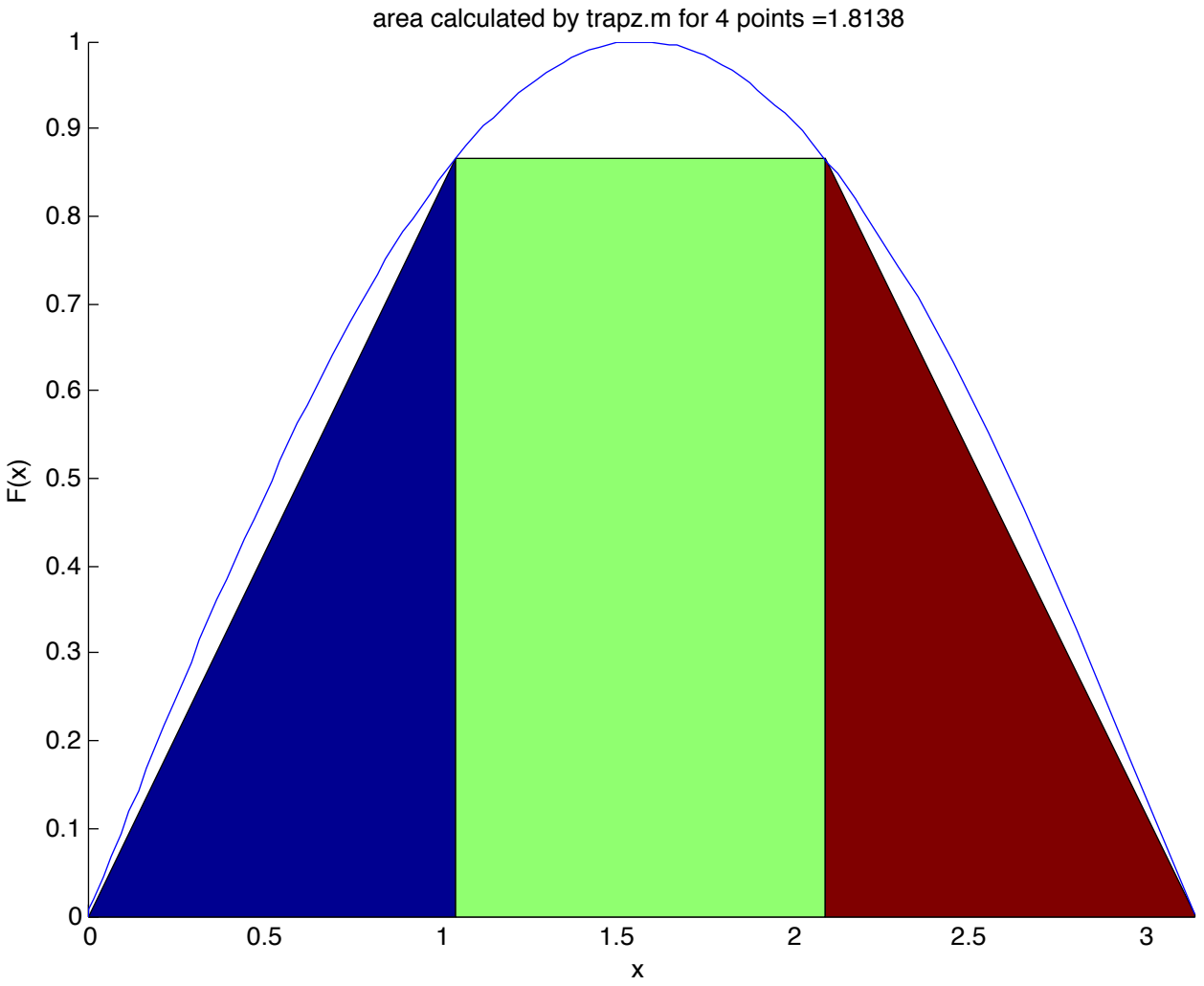
```
Trapezoid method (Method B)
np= 3
```



→ Are these rectangles? Why not?

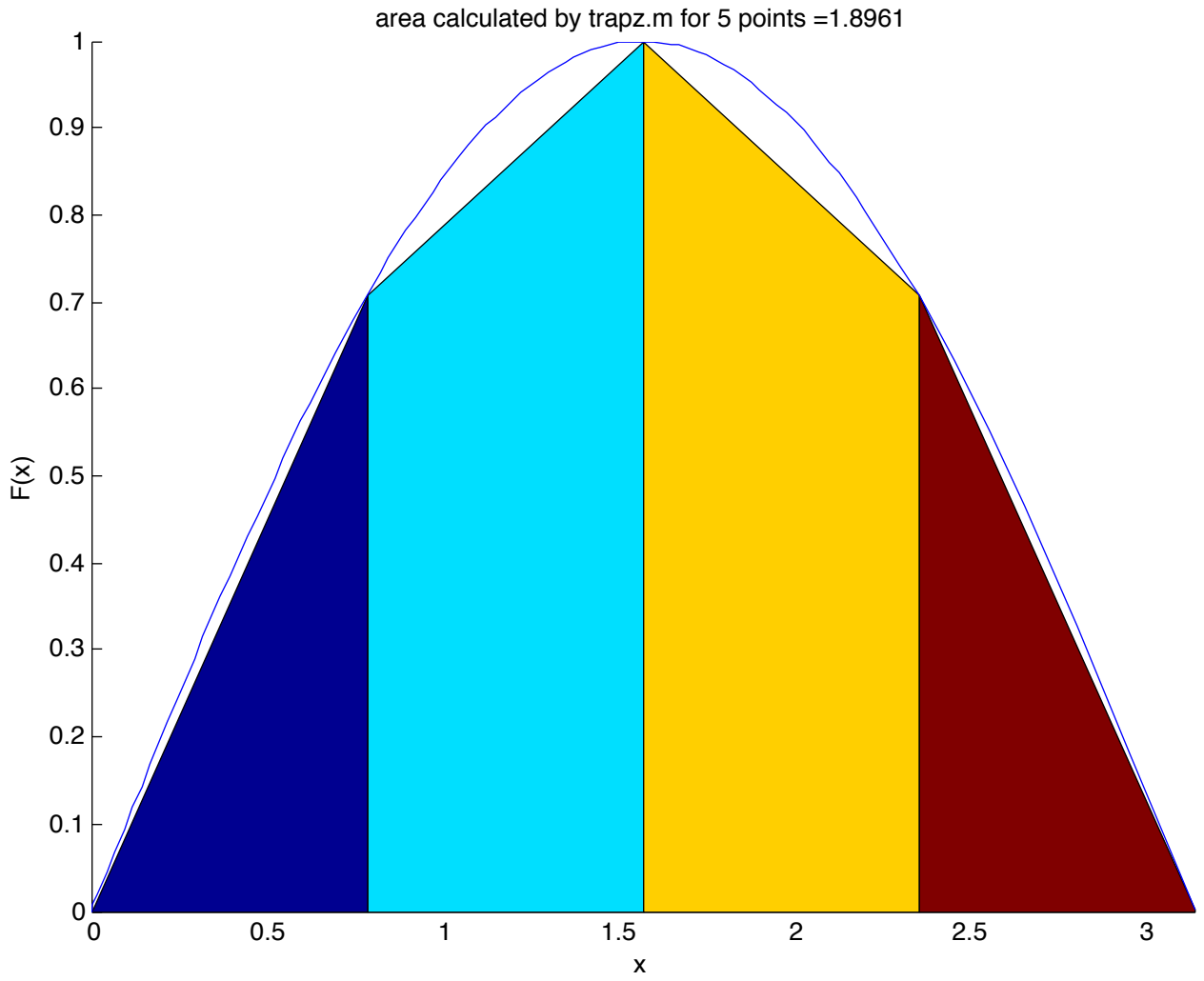
→ Three points means how many ‘rectangles’?

Trapezoid method (Method B)
np= 4

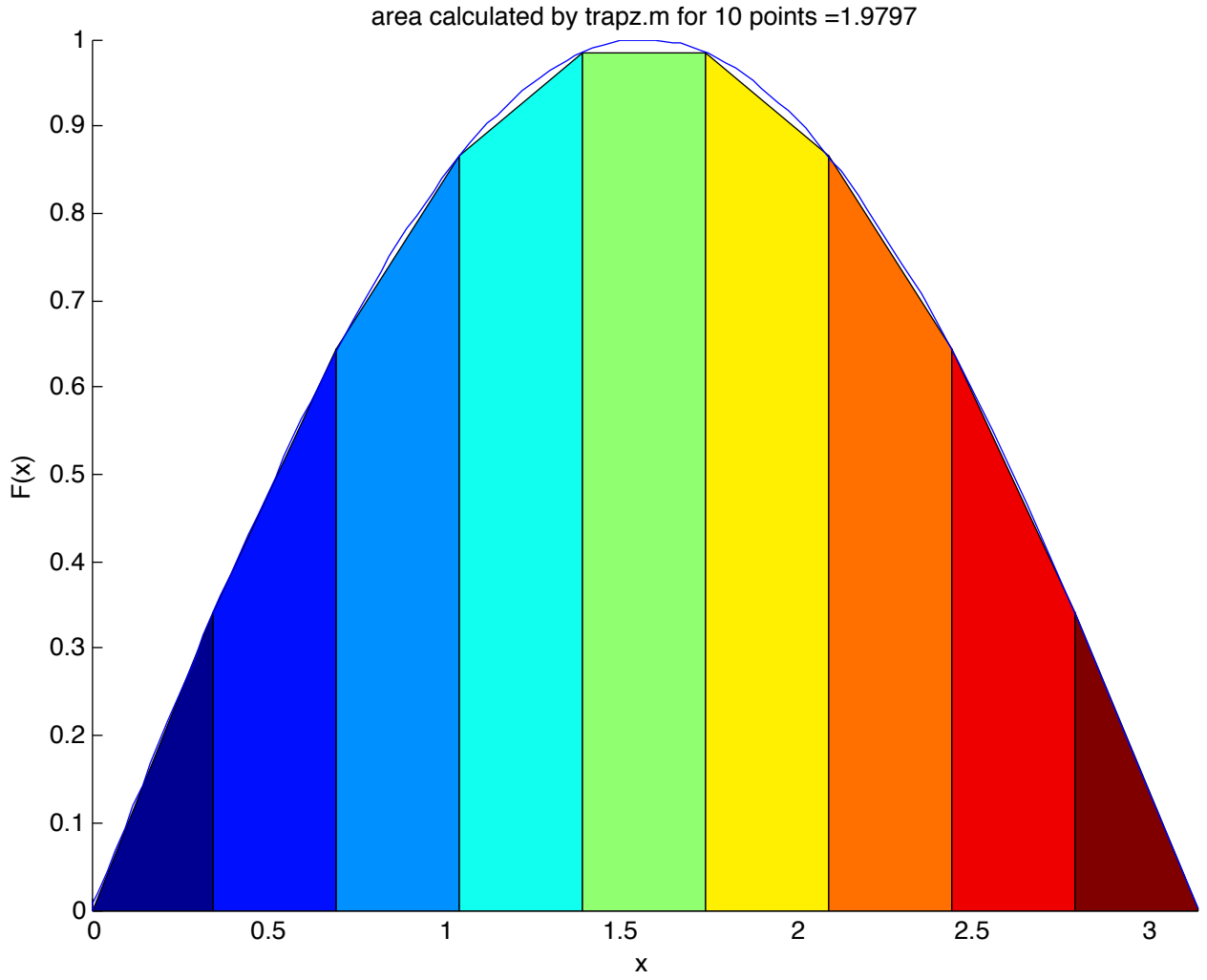


→ What is the associated ‘error’?

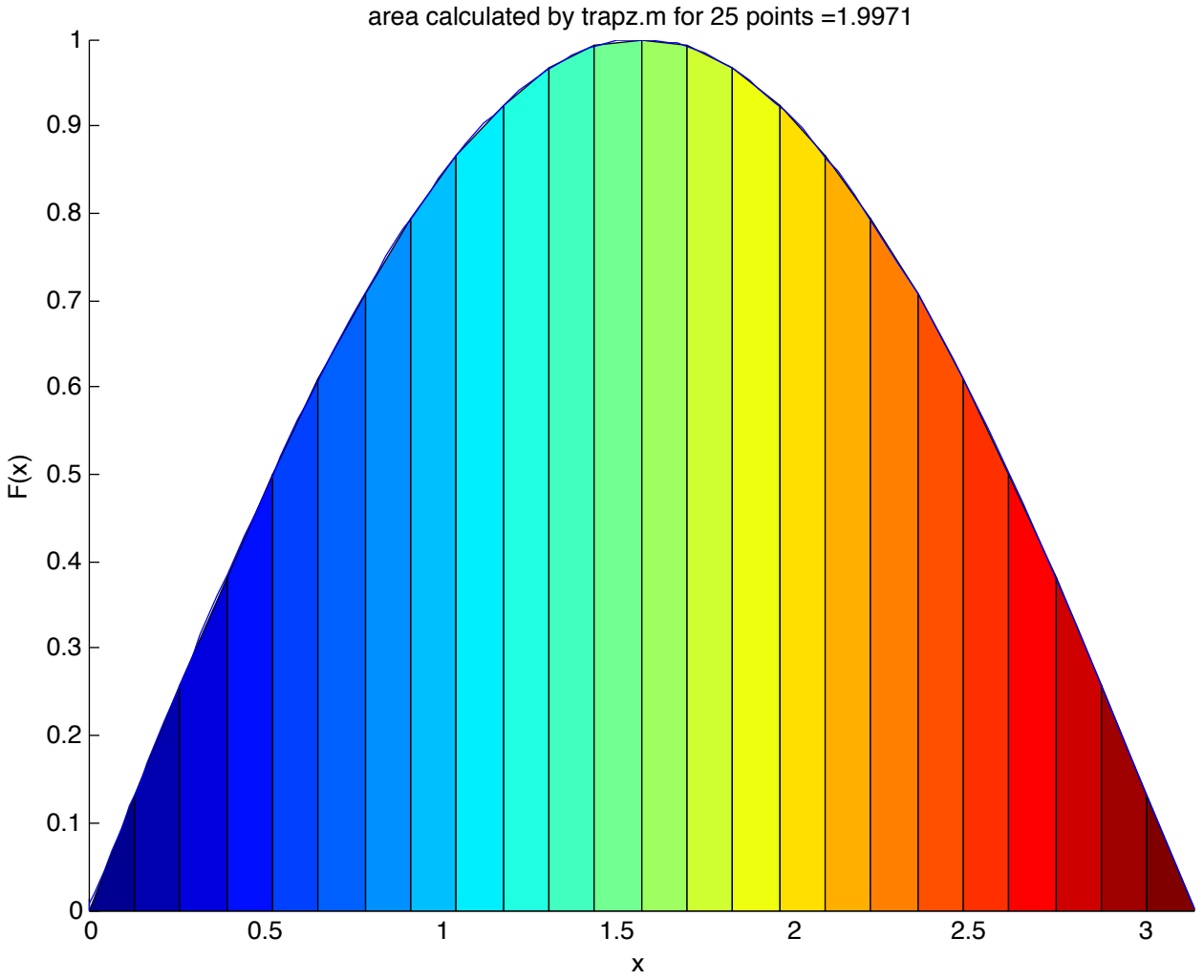
Trapezoid method (Method B)
np= 5



Trapezoid method (Method B)
np= 10



Trapezoid method (Method B)
np= 25



```
>> INTexample1
left-hand rule yields =1.9338 (for 5 steps)
right-hand rule yields = 1.93377
area calculated by trapz.m for 3 points =1.5708
area calculated by trapz.m for 4 points =1.8138
area calculated by trapz.m for 5 points =1.8961
area calculated by trapz.m for 10 points =1.9797
area calculated by trapz.m for 25 points =1.9971
area calculated by quad.m = 1.999999996
>>
```

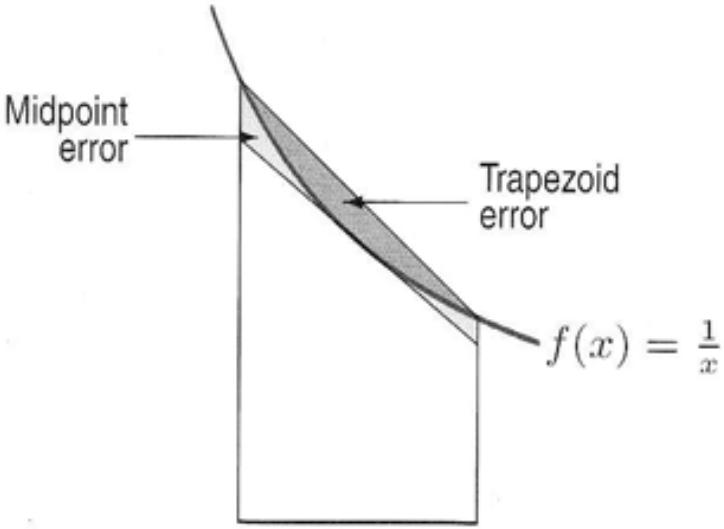
→ Why do the LEFT and RIGHT Riemann sums yield the same values?

→ Are LEFT and RIGHT better than TRAP?

→ How many ‘points’ does quad.m use?

Summary

$$\int_a^b f(x)dx \approx \sum_{j=0}^{N} f(x_j)h$$



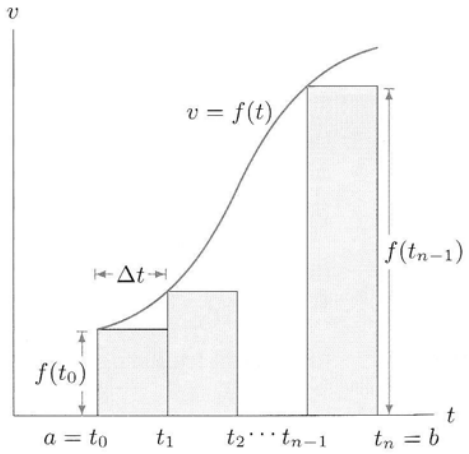


Figure 5.8: Left-hand sums

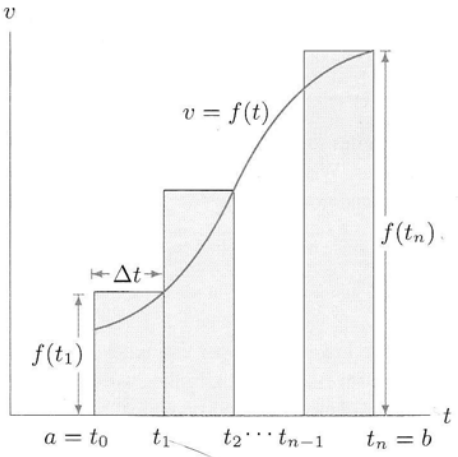


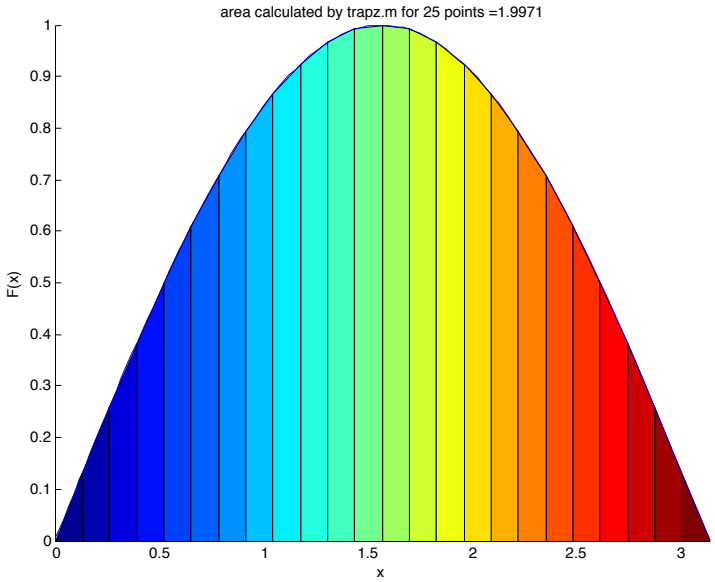
Figure 5.9: Right-hand sums

Trapezoid rule $\int_{x_0}^{x_1} f(x)dx = \frac{h}{2}(f_0 + f_1) - \frac{h^3}{12}f''(c)$ (4.2.6a)

Simpson's rule $\int_{x_0}^{x_2} f(x)dx = \frac{h}{3}(f_0 + 4f_1 + f_2) - \frac{h^5}{90}f''''(c)$ (4.2.6b)

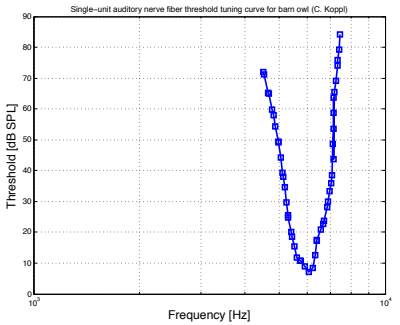
Simpson's 3/8 rule $\int_{x_0}^{x_3} f(x)dx = \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3) - \frac{3h^5}{80}f''''(c)$ (4.2.6c)

Boole's rule $\int_{x_0}^{x_4} f(x)dx = \frac{2h}{45}(7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4) - \frac{8h^7}{945}f^{(6)}(c).$ (4.2.6d)



Post-class exercises

- Determine what trapz.m and quad.m do, and what is different between them
- Write your own code version to utilize the trapezoid method
- Verify errors behave as expected (e.g., concavity)
- Calculate Qerb for the barn owl auditory nerve fiber



Directions:

1. Read in the data
2. Determine the characteristic frequency (CF) by finding where the minimum is
3. Convert the level curve from dB SPL to pascals
4. Flip the level curve over and normalize relative to CF (i.e., unity at CF, decreases to 0 on either side)
5. Integrate to get ERB
6. Qerb= CF/ERB