Part I

Single reaction

Equilibrium binding and Cooperativity

Week 1-2,

Matlab crash course

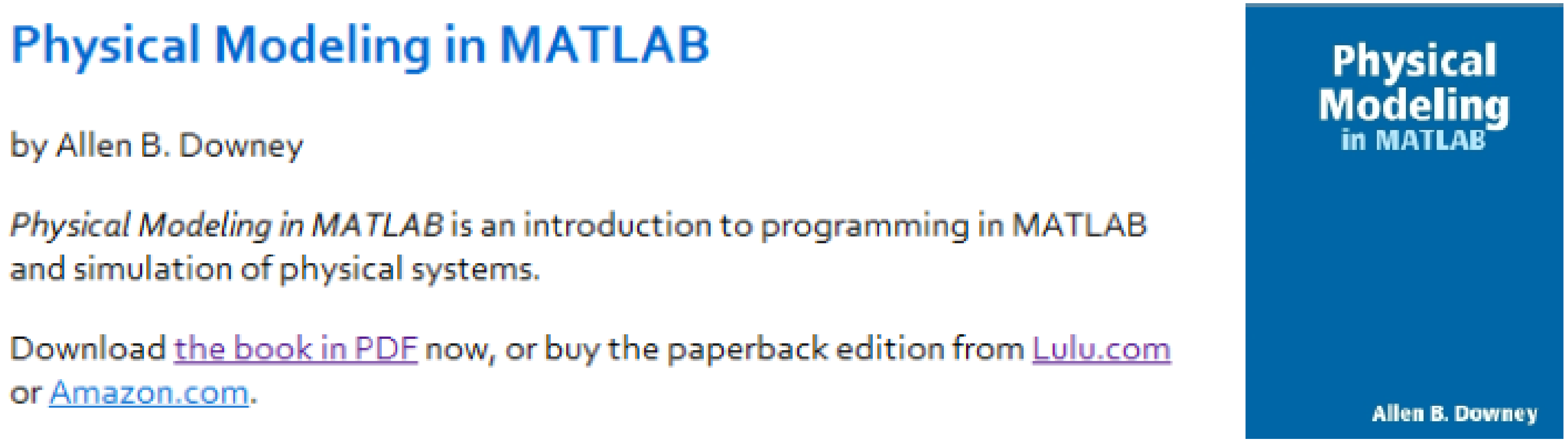
Today's objective

Get comfortable playing with Matlab…

- Interacting with Matlab
- Enter Data
- Operations
- Some Commonly Used Functions
- Making Pretty Pictures
- M-Files and Scripts
- For, While, and If
- Solving ODEs

Useful Resources

- http://www.greenteapress.com/matlab/



Physical Modeling in MATLAB

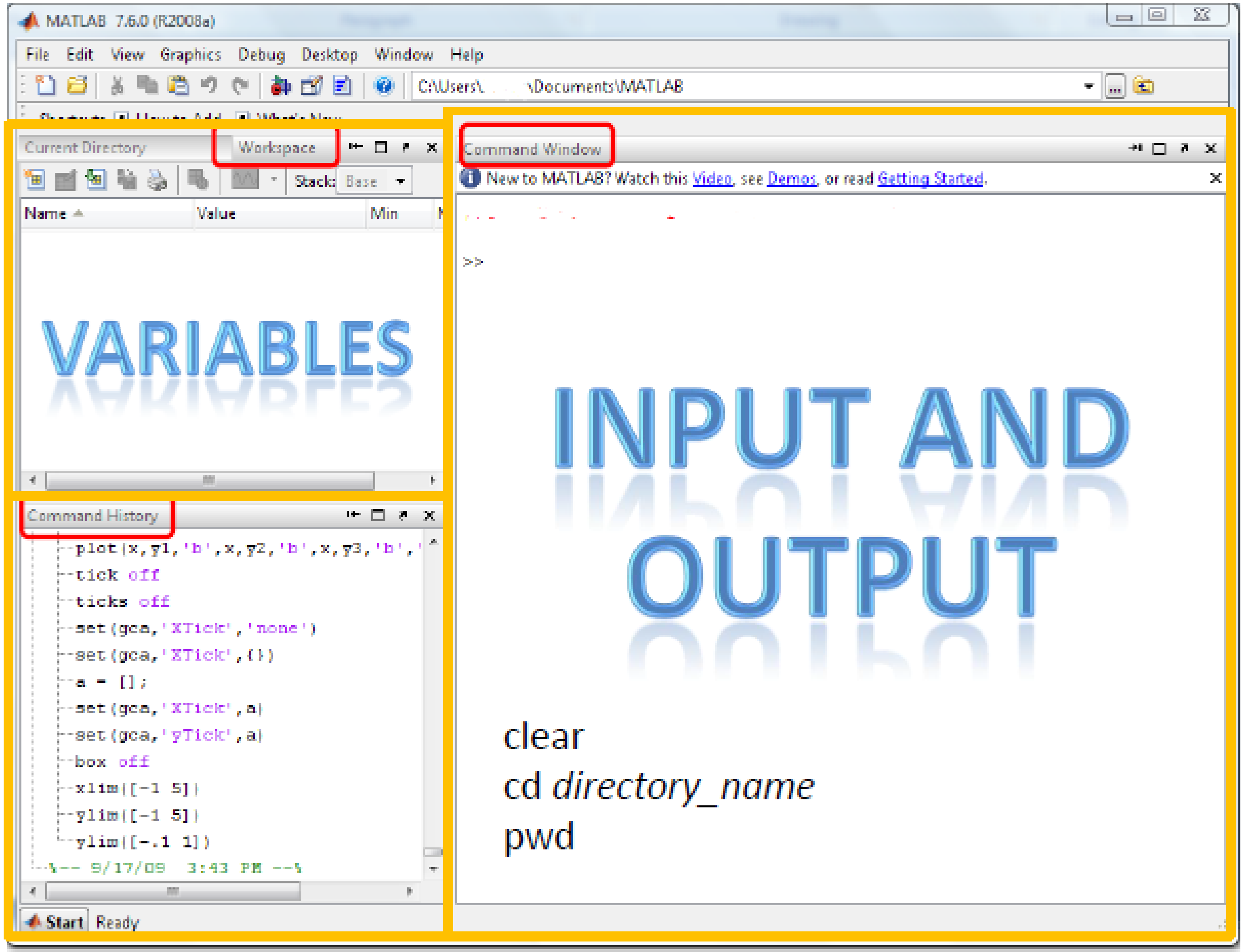
by Allen B. Downey

Physical Modeling in MATLAB is an introduction to programming in MATLAB and simulation of physical systems.

Download the book in PDF now, or buy the paperback edition from Lulu.com or Amazon.com.

- Google
  - Search for ‘matlab plot’

Interacting with Matlab


MATLAB 7.6.0 (R2008a)
File Edit View Graphics Debug Desktop Window Help
C:\Users\ \Documents\MATLAB
Current Directory
Workspace
Stack: Base
Name Value Min
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start Ready

```
>>
```

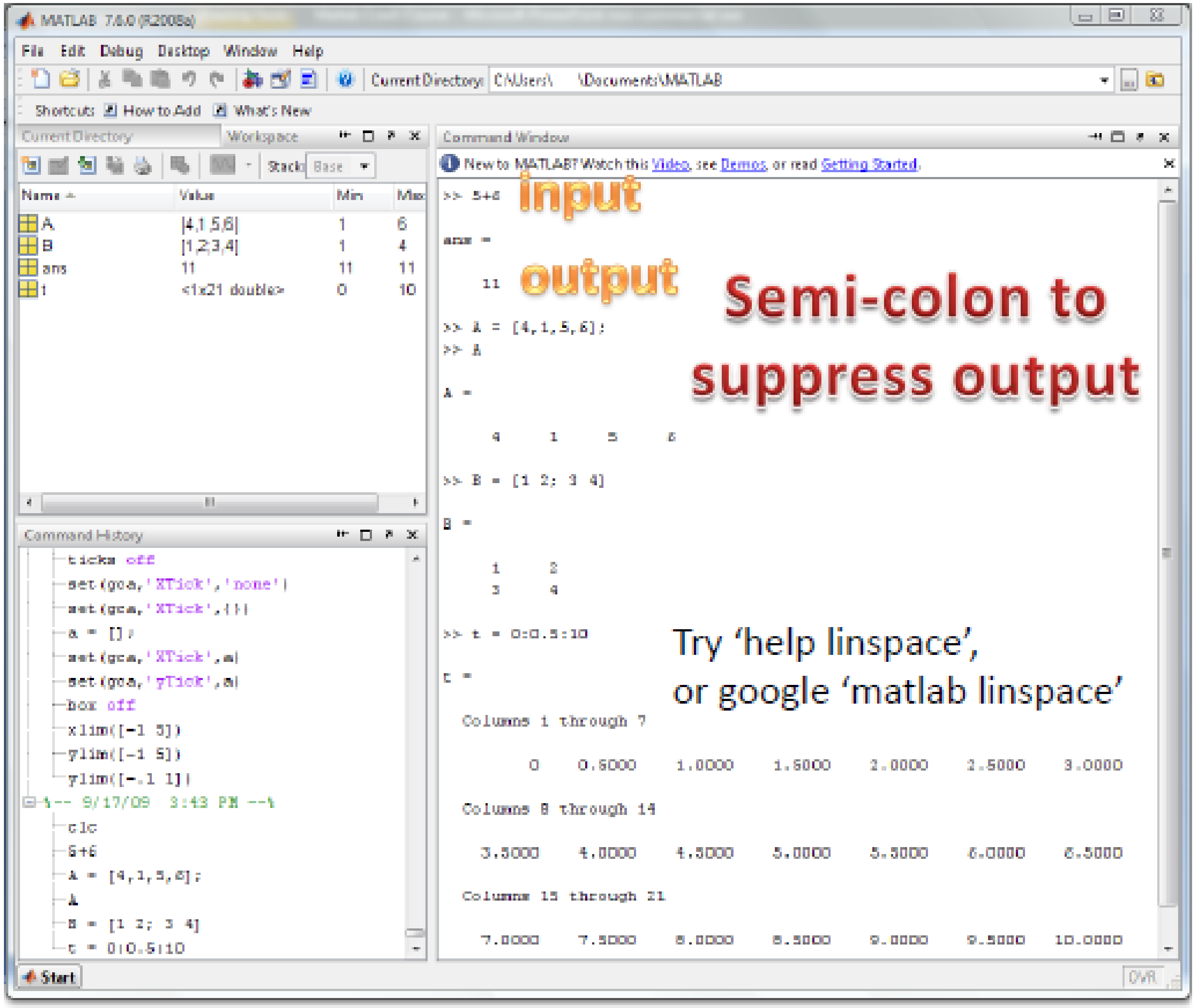
```
plot(x,y1,'b',x,y2,'b',x,y3,'b','
tick off
ticks off
set(gca,'XTick','none')
set(gca,'XTick',{})
a = [];
set(gca,'XTick',a)
set(gca,'yTick',a)
box off
xlim([-1 5])
ylim([-1 5])
ylim([-.1 1])
%-- 9/17/09  3:43 PM --%
```

VARIABLES

INPUT AND OUTPUT

clear
cd *directory_name*
pwd

Entering Data


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: C:\Users\ \Documents\MATLAB
Shortcuts How to Add What's New
Current Directory Workspace
Stack: Base
Command History
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Start
OVR

Name	Value	Min	Max
A	[4,1,5,6]	1	6
B	[1,2;3,4]	1	4
ans	11	11	11
t	<1x21 double>	0	10

```
  ticks off
  set(gca,'XTick','none')
  set(gca,'XTick',{})
  a = [];
  set(gca,'XTick',a)
  set(gca,'yTick',a)
  box off
  xlim([-1 5])
  ylim([-1 5])
  ylim([-.1 1])
%-- 9/17/09  3:43 PM --%
  clc
  5+6
  A = [4,1,5,6];
  A
  B = [1 2; 3 4]
  t = 0:0.5:10
```

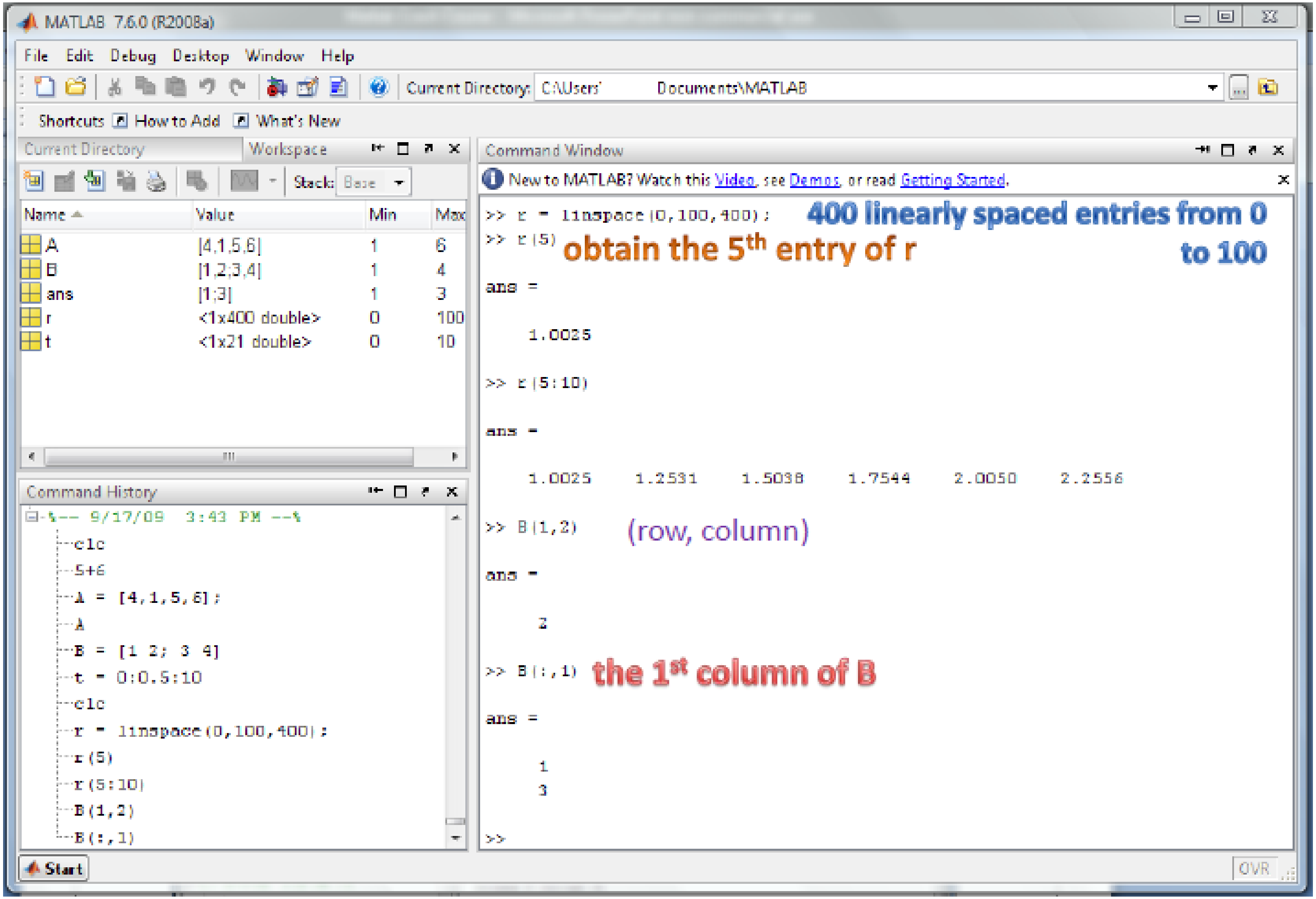
```
>> 5+6

ans =

    11

>> A = [4,1,5,6];
>> A

A =

     4     1     5     6

>> B = [1 2; 3 4]

B =

     1     2
     3     4

>> t = 0:0.5:10

t =

  Columns 1 through 7

         0    0.5000    1.0000    1.5000    2.0000    2.5000    3.0000

  Columns 8 through 14

    3.5000    4.0000    4.5000    5.0000    5.5000    6.0000    6.5000

  Columns 15 through 21

    7.0000    7.5000    8.0000    8.5000    9.0000    9.5000   10.0000
```

input

output

Semi-colon to suppress output

Try 'help linspace',
or google 'matlab linspace'


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: C:\Users\ Documents\MATLAB
Shortcuts How to Add What's New
Current Directory Workspace
Stack: Base
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start
OVR

Name	Value	Min	Max
A	[4,1,5,6]	1	6
B	[1,2;3,4]	1	4
ans	[1;3]	1	3
r	<1x400 double>	0	100
t	<1x21 double>	0	10

```
%-- 9/17/09  3:43 PM --%
clc
5+6
A = [4,1,5,6];
A
B = [1 2; 3 4]
t = 0:0.5:10
clc
r = linspace(0,100,400);
r(5)
r(5:10)
B(1,2)
B(:,1)
```

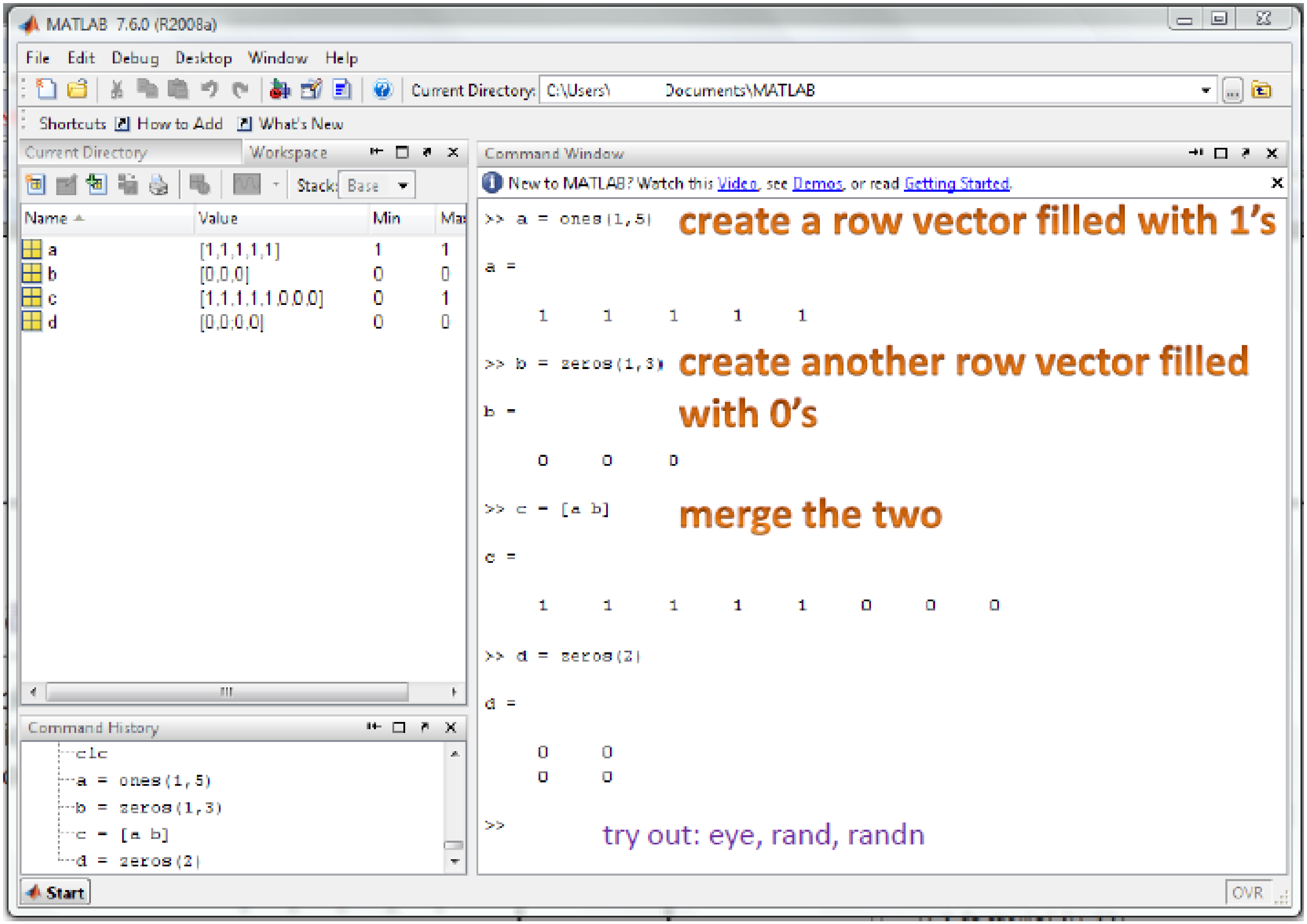
```
>> r = linspace(0,100,400);
>> r(5)

ans =

    1.0025

>> r(5:10)

ans =

    1.0025    1.2531    1.5038    1.7544    2.0050    2.2556

>> B(1,2)

ans =

     2

>> B(:,1)

ans =

     1
     3

>>
```

400 linearly spaced entries from 0 to 100

obtain the 5th entry of r

(row, column)

the 1st column of B



MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: C:\Users\ Documents\MATLAB
Shortcuts How to Add What's New
Current Directory Workspace
Stack: Base
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start
OVR

Name	Value	Min	Max
a	[1,1,1,1,1]	1	1
b	[0,0,0]	0	0
c	[1,1,1,1,1,0,0,0]	0	1
d	[0,0;0,0]	0	0

```
>> a = ones(1,5)

a =

     1     1     1     1     1

>> b = zeros(1,3)

b =

     0     0     0

>> c = [a b]

c =

     1     1     1     1     1     0     0     0

>> d = zeros(2)

d =

     0     0
     0     0

>>
```

create a row vector filled with 1's

create another row vector filled with 0's

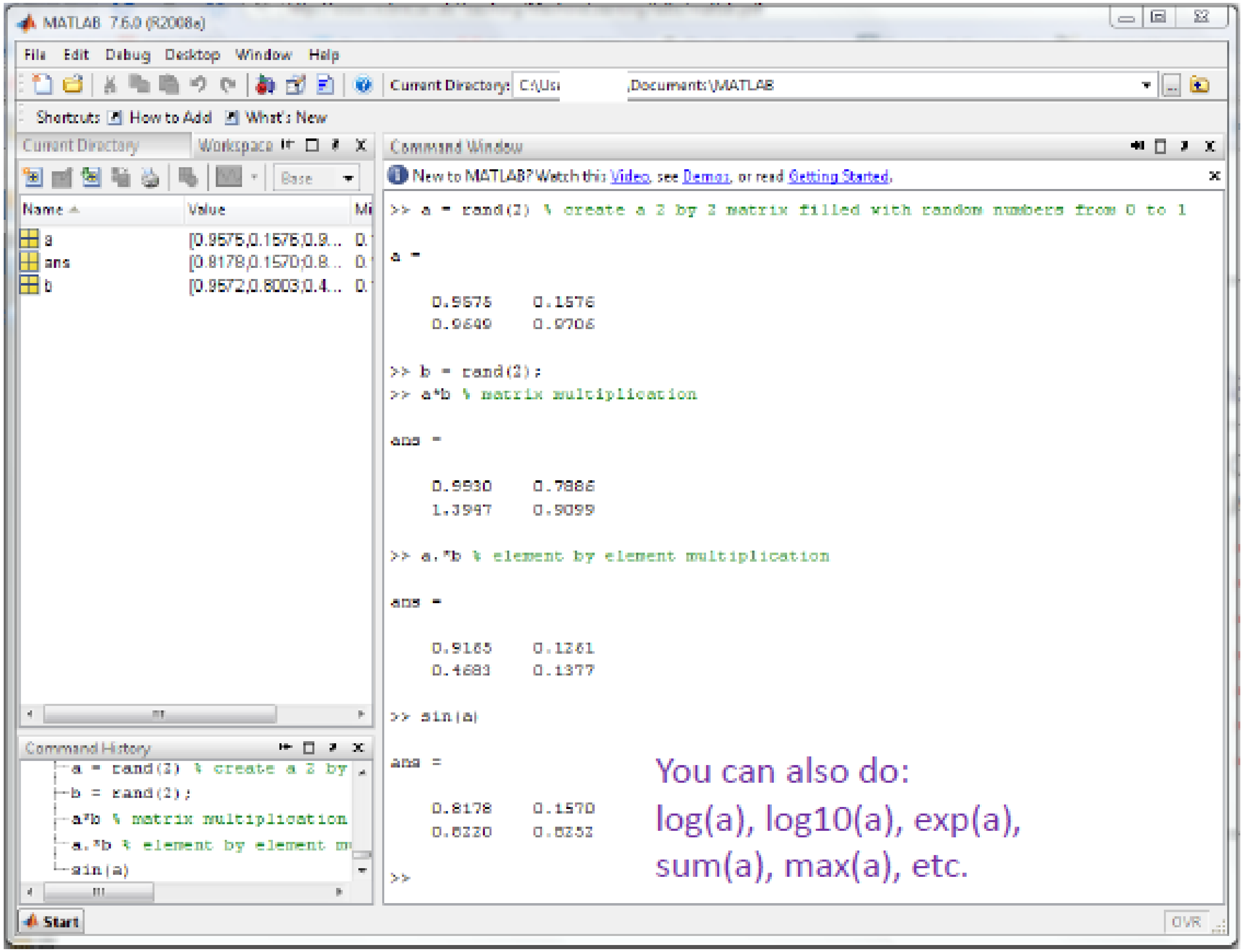
merge the two

try out: eye, rand, randn

Command History:

```
clc
a = ones(1,5)
b = zeros(1,3)
c = [a b]
d = zeros(2)
```

Operations


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: C:\Us ,Documents\MATLAB
Shortcuts How to Add What's New
Current Directory Workspace
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start
OVR

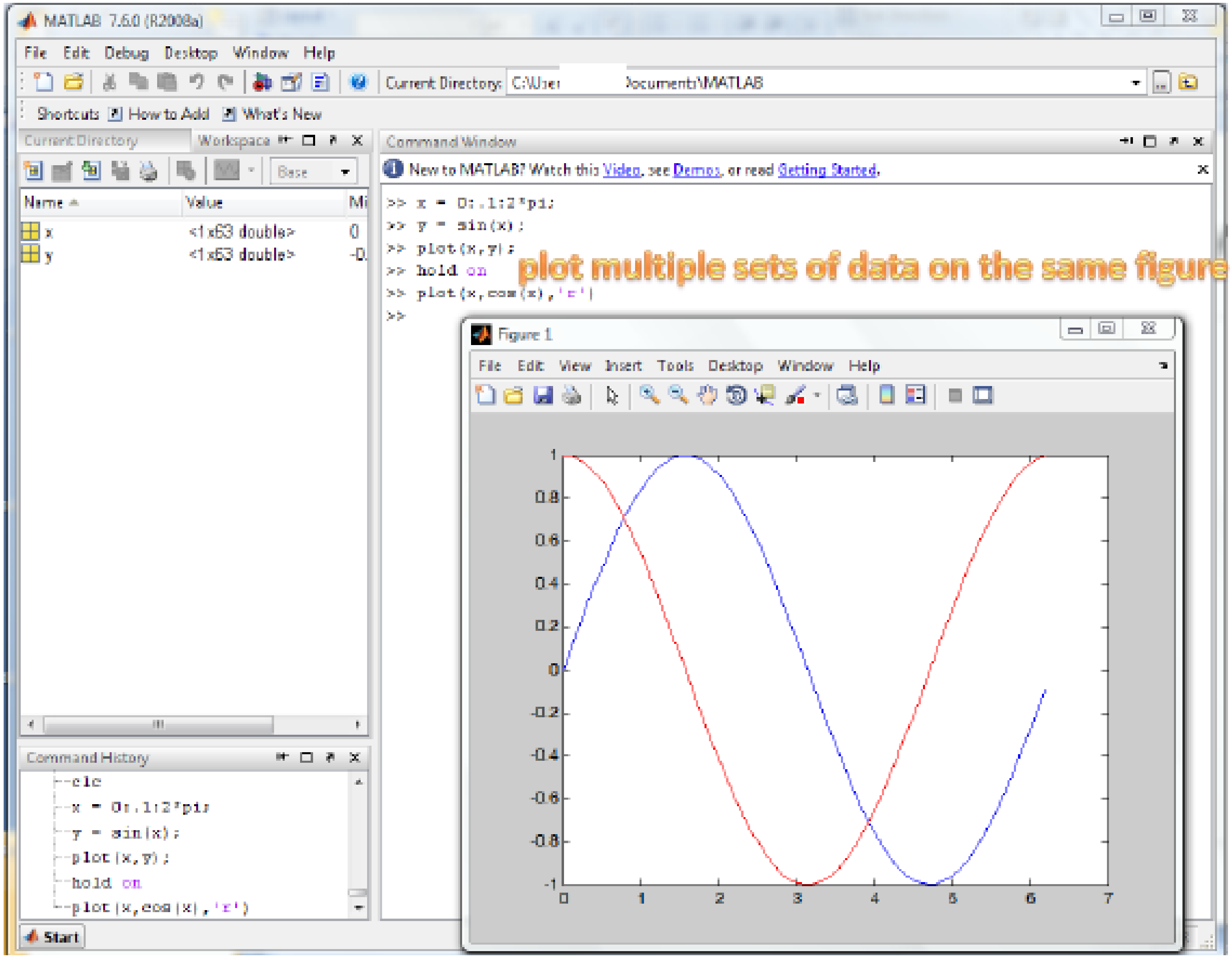
Name	Value
a	[0.9575,0.1576;0.9...
ans	[0.8178,0.1570;0.8...
b	[0.9572,0.8003;0.4...

```
>> a = rand(2) % create a 2 by 2 matrix filled with random numbers from 0 to 1

a =

    0.9575    0.1576
    0.9649    0.9706

>> b = rand(2);
>> a*b % matrix multiplication

ans =

    0.9930    0.7886
    1.3947    0.9099

>> a.*b % element by element multiplication

ans =

    0.9165    0.1261
    0.4683    0.1377

>> sin(a)

ans =

    0.8178    0.1570
    0.8220    0.8252

>>
```

```
a = rand(2) % create a 2 by
b = rand(2);
a*b % matrix multiplication
a.*b % element by element mu
sin(a)
```

You can also do:
log(a), log10(a), exp(a),
sum(a), max(a), etc.

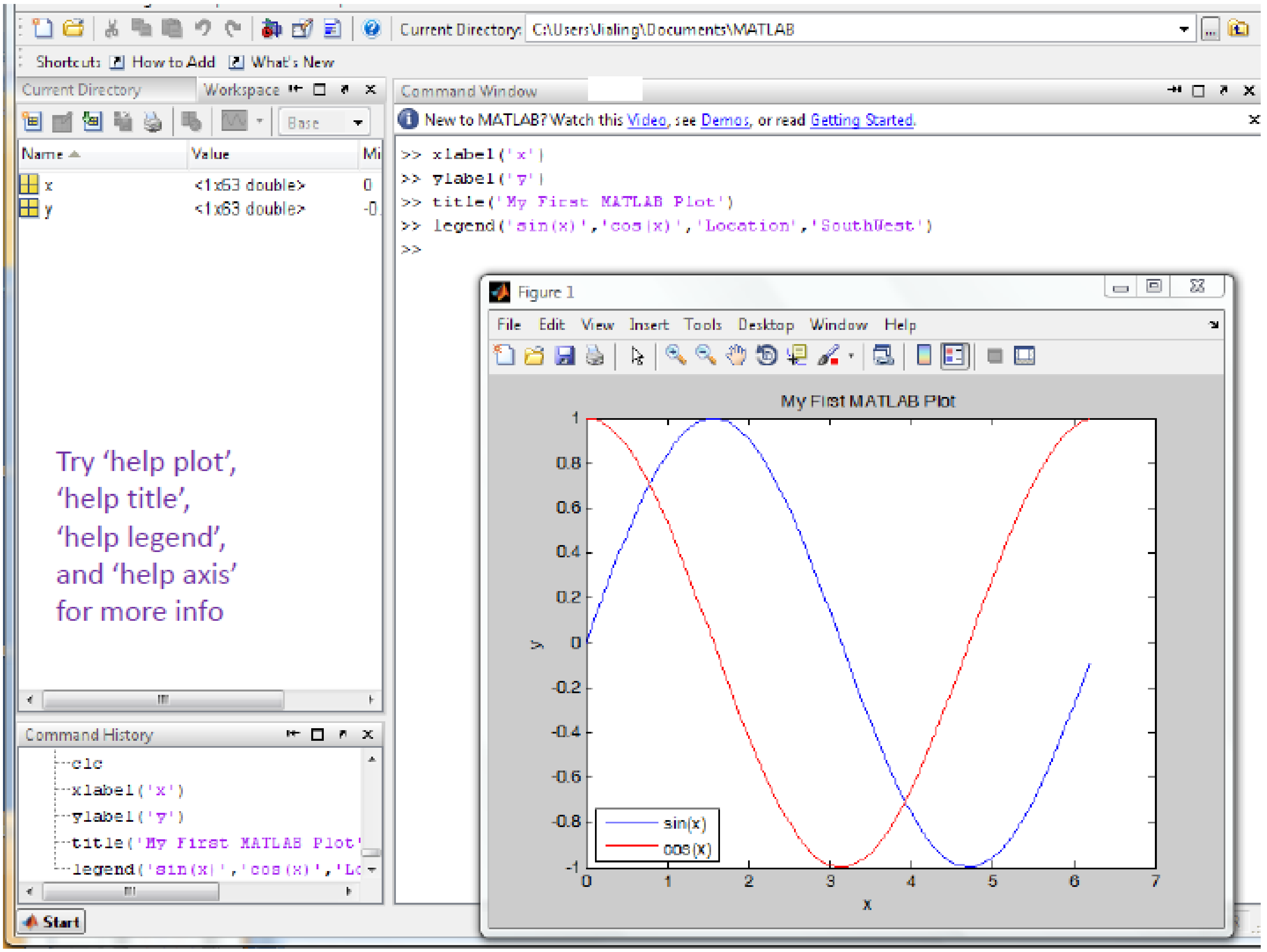
Plotting Data / Making Pretty Pictures



```
>> x = 0:.1:2*pi;
>> y = sin(x);
>> plot(x,y);
>> hold on
>> plot(x,cos(x),'r')
>>
```

plot multiple sets of data on the same figure

MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: C:\Users\ Documents\MATLAB
Shortcuts How to Add What's New
Current Directory Workspace
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Name Value
x <1x63 double>
y <1x63 double>
Command History
clc
x = 0:.1:2*pi;
y = sin(x);
plot(x,y);
hold on
plot(x,cos(x),'r')
Start
Figure 1
File Edit View Insert Tools Desktop Window Help
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0 1 2 3 4 5 6 7



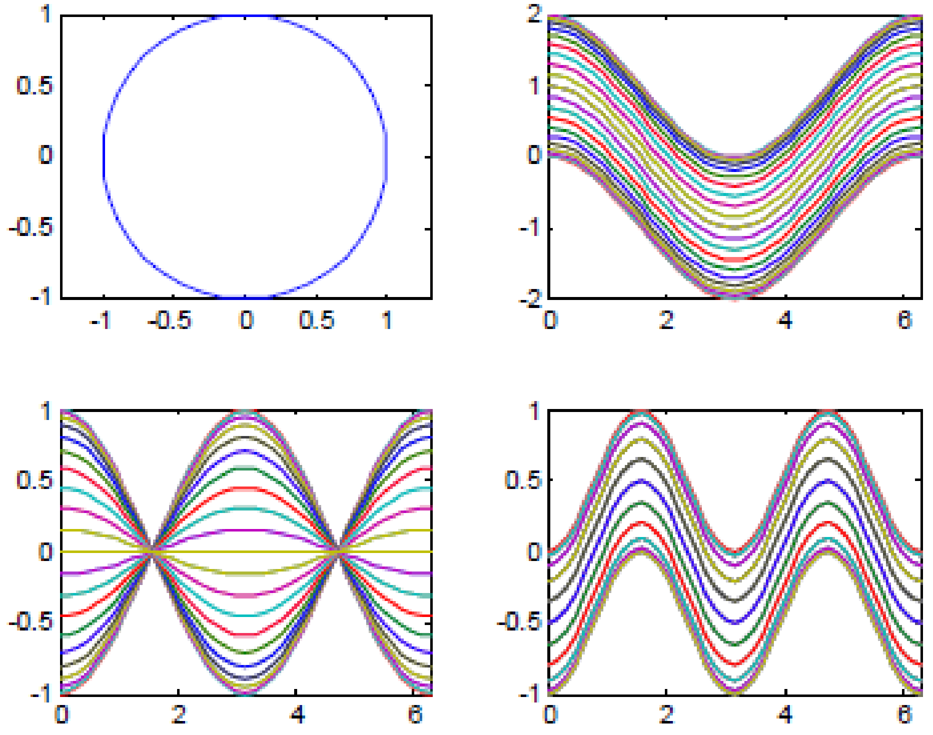
```
>> xlabel('x')
>> ylabel('y')
>> title('My First MATLAB Plot')
>> legend('sin(x)','cos(x)','Location','SouthWest')
>>
```

Current Directory: C:\Users\Jialing\Documents\MATLAB
Shortcuts How to Add What's New
Current Directory Workspace
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Name Value Mi
x <1x63 double> 0
y <1x63 double> -0.
Command History
clc
xlabel('x')
ylabel('y')
title('My First MATLAB Plot'
legend('sin(x)','cos(x)','Lc
Start
Figure 1
File Edit View Insert Tools Desktop Window Help
My First MATLAB Plot
sin(x)
cos(x)
x
y

Try 'help plot', 'help title', 'help legend', and 'help axis' for more info

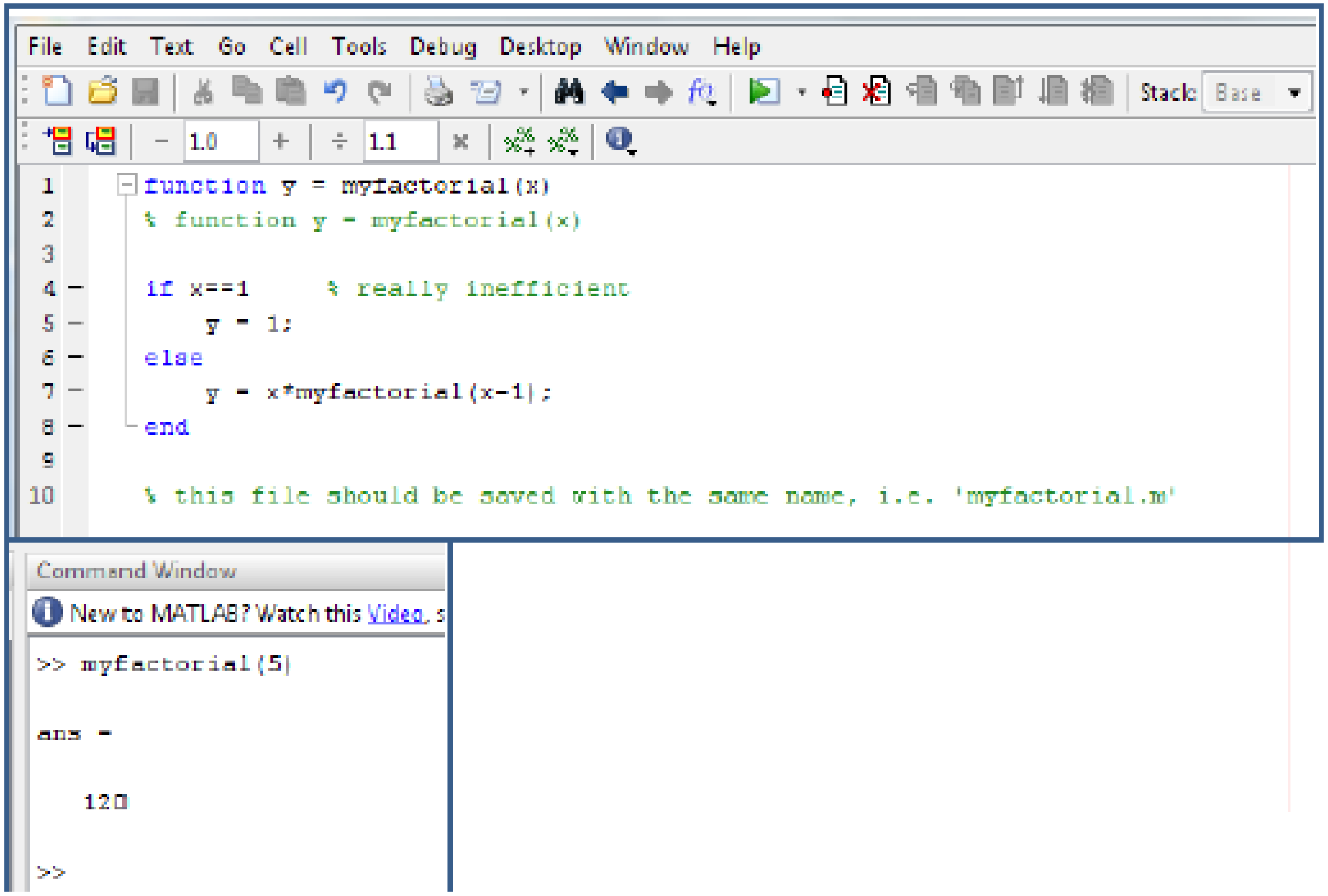
More About Plotting

```
t = 0:pi/20:2*pi;
[x,y] = meshgrid(t);   % look up meshgrid

subplot(2,2,1)    % creates a 2x2 array of plots, and plot in the first subplot
plot(sin(t),cos(t))
axis equal   % this is a parametric plot

subplot(2,2,2)
z = sin(x)+cos(y);    % z is a matrix
plot(t,z)
axis([0 2*pi -2 2])    % plotting each column of z
                       % versus t

subplot(2,2,3)
z = sin(x).*cos(y);
plot(t,z)
axis([0 2*pi -1 1])

subplot(2,2,4)
z = (sin(x).^2)-(cos(y).^2);
plot(t,z);
axis([0 2*pi -1 1])

% for 3-D plotting, try mesh, surf, surfl, waterfall, etc
```



M-Files and Functions

- Let’s make our own functions
- To start the editor, type ‘edit’

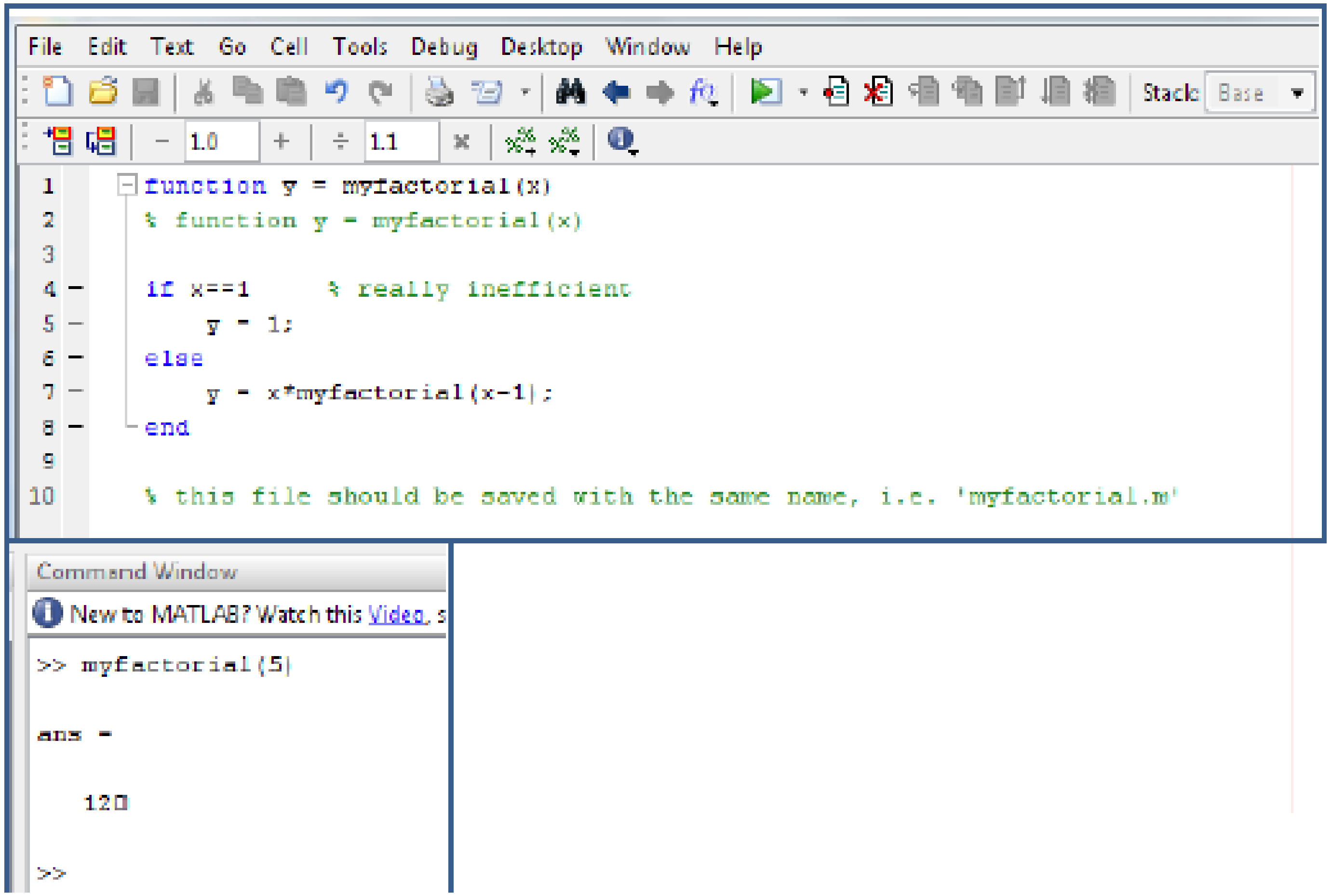

File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
1.0
1.1
Command Window
New to MATLAB? Watch this Video, s

```
function y = myfactorial(x)
% function y = myfactorial(x)

if x==1      % really inefficient
    y = 1;
else
    y = x*myfactorial(x-1);
end

% this file should be saved with the same name, i.e. 'myfactorial.m'
```

```
>> myfactorial(5)

ans =

   120

>>
```

M-Files and Functions

- Local workspace and Scoping
- To make variables global: global *variable_name*


File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
1.0
1.1
Command Window
New to MATLAB? Watch this Video, s

```
function y = myfactorial(x)
% function y = myfactorial(x)

if x==1     % really inefficient
    y = 1;
else
    y = x*myfactorial(x-1);
end

% this file should be saved with the same name, i.e. 'myfactorial.m'
```

```
>> myfactorial(5)

ans =

   120

>>
```

For, While, and If

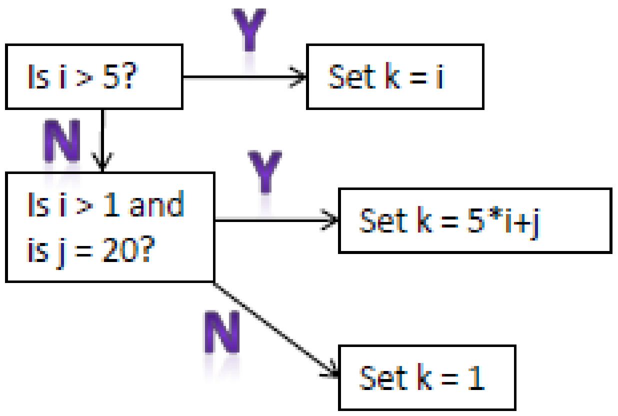
```
for m = 1:100
        num = 1/(m+1)
end
```

A for loop

```
-----------------------------
% find all the powers
% of 2 below 10000
while num < 10000
        num = 2^i;
        v = [v; num];
        i = i+1;
end
```

A while loop

```
-----------------------------
i = 6; j = 21;
if i > 5
        k = i;
elseif (i > 1) & (j == 20)
        k = 5*i+j;
else
        k = 1;
end
```



• And:	a & b
• Or:	a | b
• Not-equal:	a ~=b
• Equal:	a == b

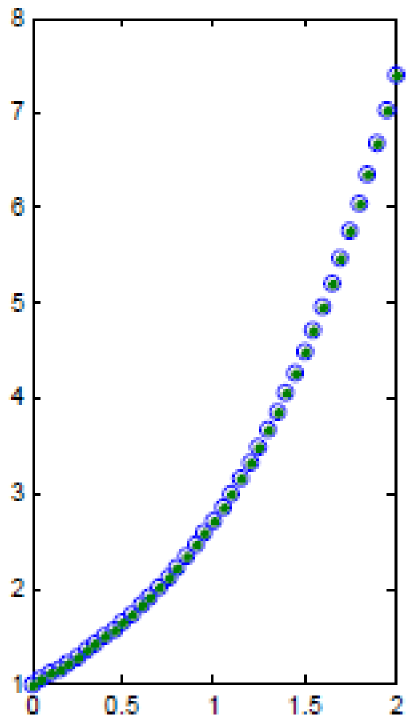
Solving ODEs

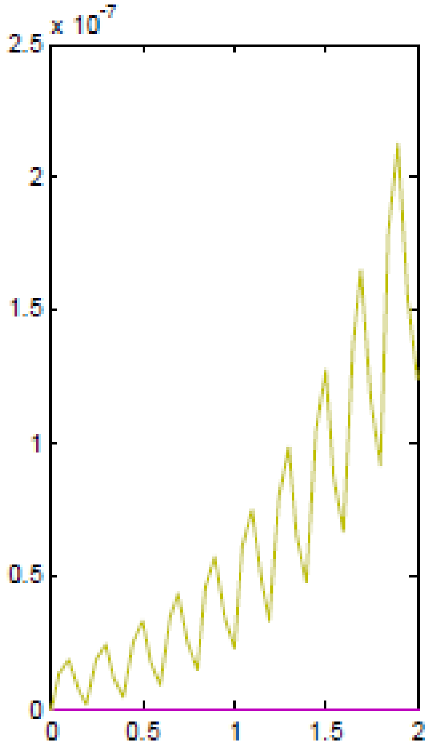
- A very simple case: $\dfrac{dy}{dt} = y(t) \qquad 0 \le t \le 2 \qquad y(0) = 1$

```
function dy = simpleode(t,y)
dy = y;  % save as simpleode.m
```

- Type in command line:

```
[t y] = ode45(@simpleode, [0, 2], [1]);
subplot(1,2,1),plot (t,y,'o',t,exp(t),'.')
subplot(1,2,2),plot(t,(y-exp(t))/exp(t))
```





Solving ODEs

Lorenz equations $\frac{dx}{dt} = c(y - x), \frac{dy}{dt} = x(r - z) - y, \frac{dz}{dt} = xy - bz$

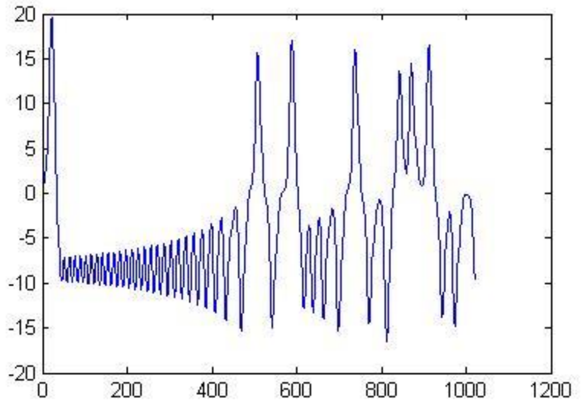
$c = 10, r = 28, b = 8/3, x(0) = y(0) = z(0) = 1, 0 < t < 30$

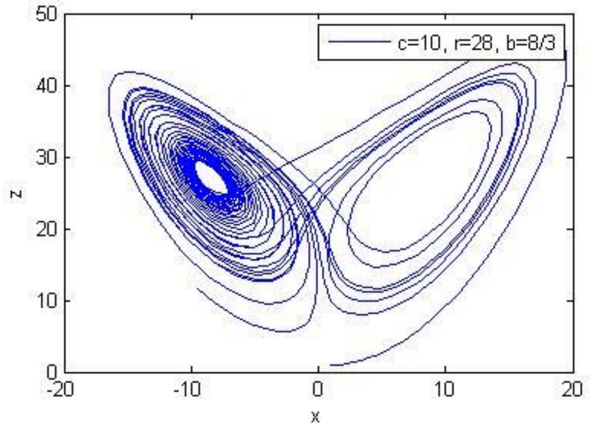
Edit and save as lorenzfunc.m

```
function dydt=f(t,y,flag,c,r,b)
% x=y(1), y=y(2),z=y(3)
dydt=[c*(y(2)-y(1)); y(1)*(r-y(3))-y(2);
                      y(1)*y(2)-b*y(3)];
```

Edit and save as lorenz.m

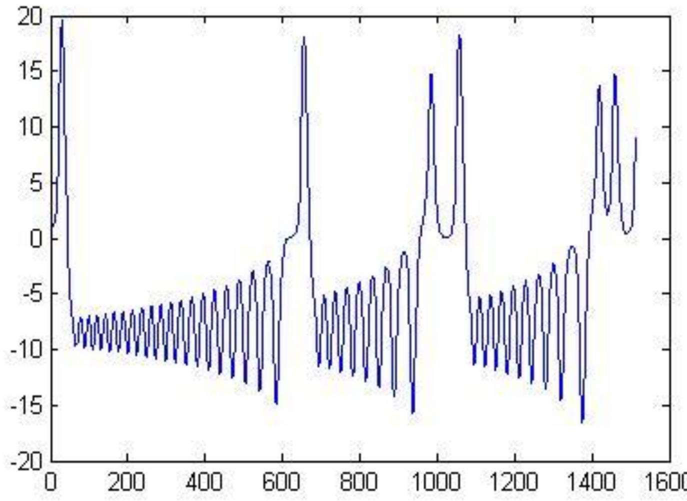
```
clear;
c=10; r=28; b=8/3;
options=[];
x0=[1 1 1];
[t y]=ode45(‘lorenzfunc’,[0 30], x0, options, c,r,b);
subplot(2,1,1);plot(y(:,1));
subplot(2,1,2);plot(y(:,1),y(:,3));
xlabel('x'); ylabel('z');
legend('c=10, r=28, b=8/3');
```





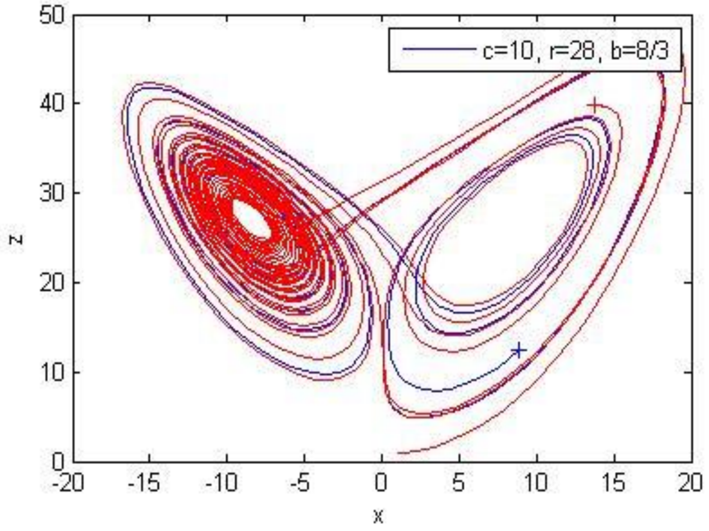
Bufferfly effect

Exponential amplification of small errors!



Add to the end of lorenz.m

```
hold on;
plot(y(end,1),y(end,3),'b+');

x1=x0+[0 0 1e-6];
[t1 y1]=ode45('lorenzfunc',[0 30],x1,options,c,r,b);
subplot(2,1,2); hold on
plot(y1(:,1),y1(:,3),'r-');
plot(y1(end,1),y1(end,3),'r+');
hold off
```



Review of Michaelis-Menten Kinetics

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_2} E + P$$

$$\frac{d[S]}{dt} = -k_1[E][S] + k_{-1}[ES]$$

$$\frac{d[E]}{dt} = -k_1[E][S] + (k_{-1} + k_2)[ES]$$

$$\frac{d[ES]}{dt} = k_1[E][S] - (k_{-1} + k_2)[ES]$$

$$\frac{d[P]}{dt} = k_2[ES] \equiv v$$

Simplified Equations

$$E_o = [E] + [ES]$$

$$\frac{d[S]}{dt} = -k_1 E_o [S] + (k_1 [S] + k_{-1})[ES]$$

$$\frac{d[ES]}{dt} = k_1 E_o [S] - (k_1 [S] + k_{-1} + k_2)[ES]$$

Initial conditions:

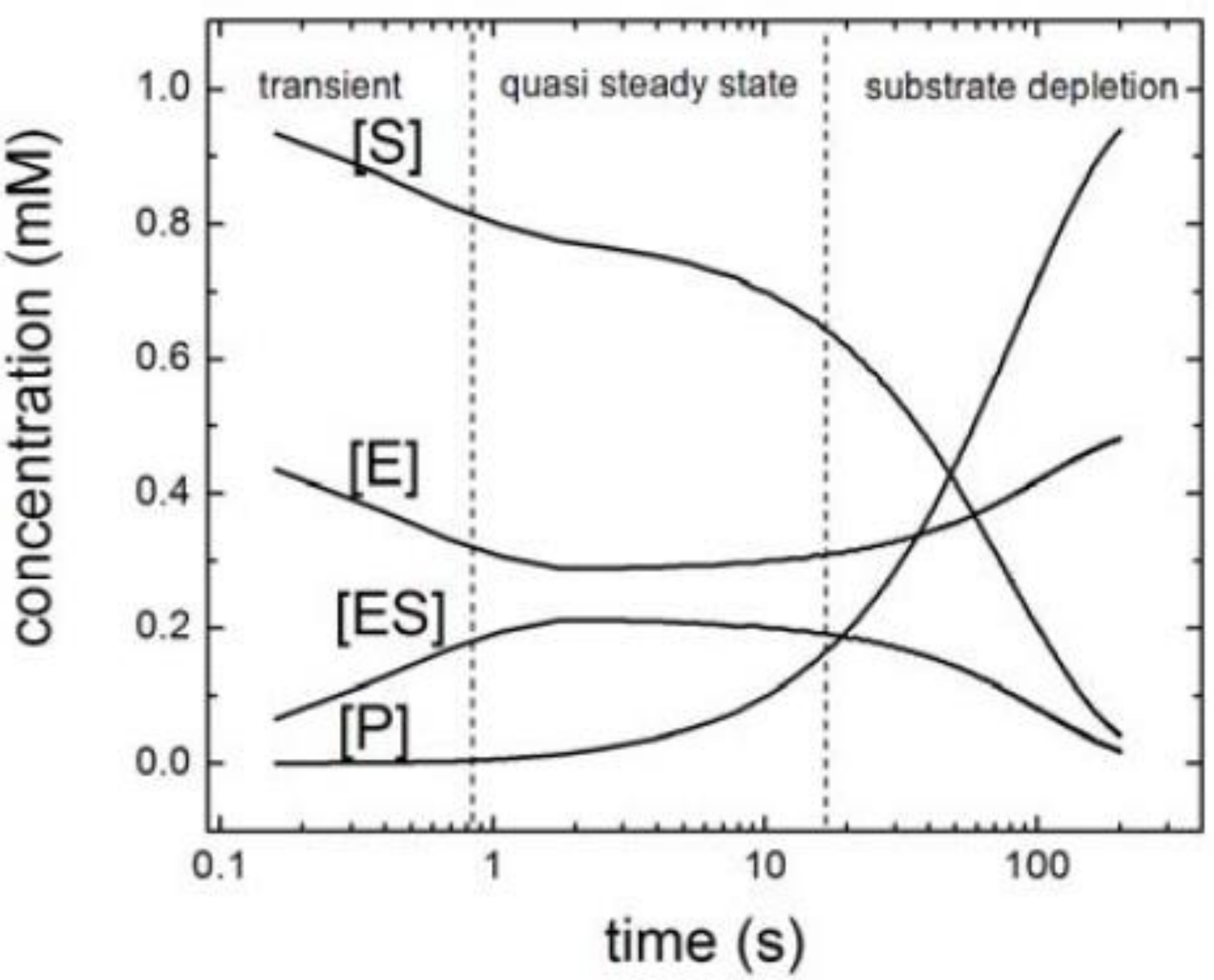
$$[S]_{t=0} = S_o$$

$$[E]_{t=0} = E_o$$

$$[ES]_{t=0} = 0$$

$$[P]_{t=0} = 0$$

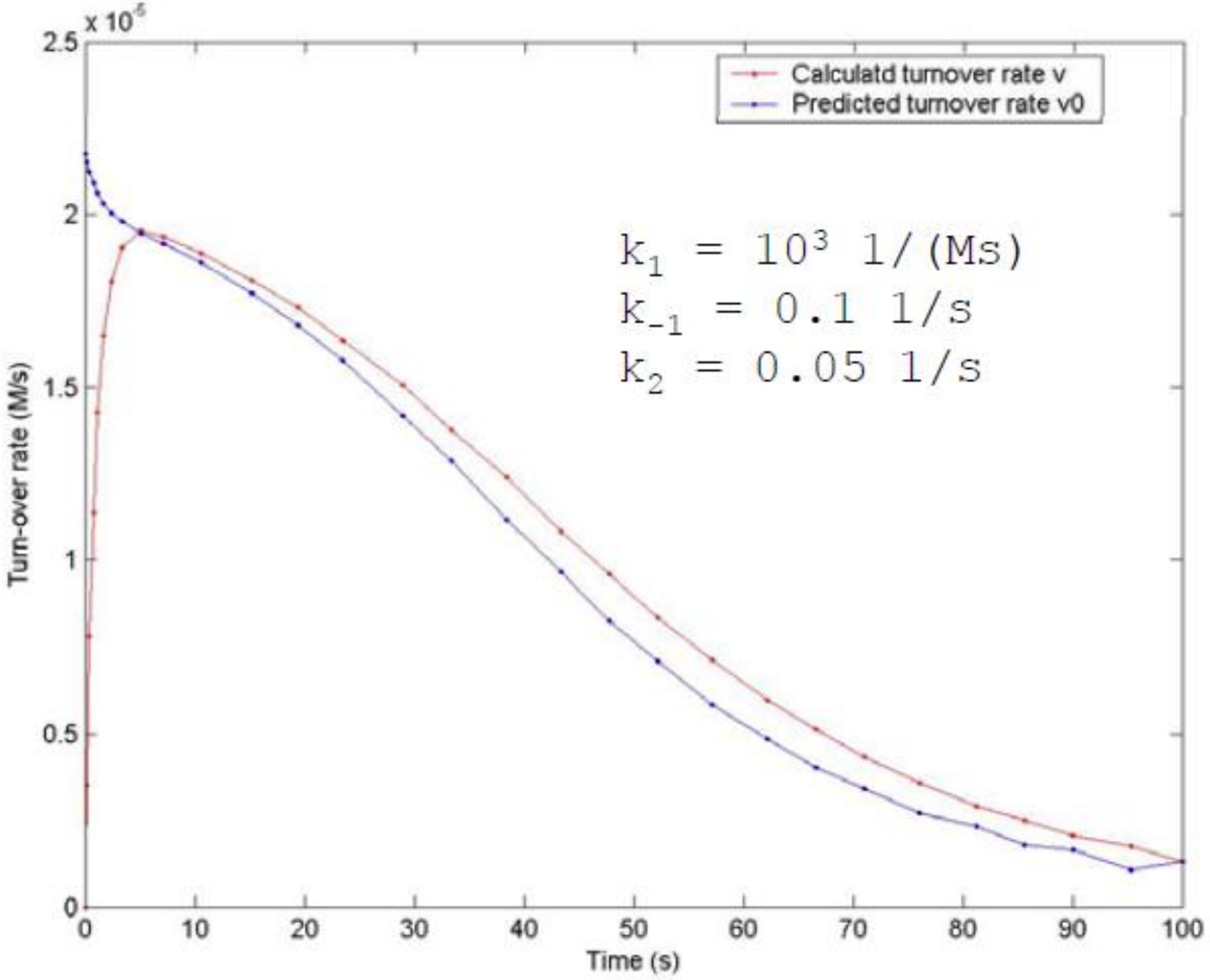
By solve the simplified equation we derive this description



Further simplification to Michaelis-Menten equation

$$v = \frac{v_{max} S}{K_m + S}$$

Good approximation if
in quasi-steady state


x 10^-5
Calculatd turnover rate v
Predicted turnover rate v0
k_1 = 10^3 1/(Ms)
k_-1 = 0.1 1/s
k_2 = 0.05 1/s
Turn-over rate (M/s)
Time (s)
0
0.5
1
1.5
2
2.5
0
10
20
30
40
50
60
70
80
90
100

Practice:
change the kinetic parameters from
k1=1e3, k-1=0.1, k2=0.05
to
k1=1e3, k-1=1e-4, k2=0.05
And compare the difference between simulation and quasi-steady state approximation again using mm.m and mmfunc.m.
And explain why.

Equilibrium binding

- Previously it was assumed that one substrate molecule binds to one enzyme
- In biological reactions proteins often bind multiple substrates
- Assume protein P has n binding sites, Pj denotes protein bound to j substrates S

$$S + P_{j-1} \leftrightarrow P_j$$

where j = 1, 2, ..., n.

- For unbound protein P_0

$$\frac{d[P_0]}{dt} = -k_{+1}[P_o][S] + k_{-1}[P_1]$$

Equilibrium binding

- In steady state, $d[P_0]/dt=0$, the association constant K_a $(1/K_d)$

$$K_a = \frac{k_{+1}}{k_{-1}} \qquad K_a = \frac{[P_1]}{[P_o][S]}$$

- To characterize all n reactions: we introduce the n association constants K_j

$$K_j = \frac{[P_j]}{[P_{j-1}][S]}$$

- For average number of substrates bound r

$$r = \frac{[P_1]+2[P_2]+3[P_3]+\ldots+n[P_n]}{[P_o]+[P_1]+[P_2]+\ldots+[P_n]}$$

$$r = \frac{K_1[S]+2K_1K_2[S]^2+3K_1K_2K_3[S]^3+\ldots+nK_1K_2\ldots K_n[S]^n}{1+K_1[S]+K_1K_2[S]^2+\ldots+K_1K_2\ldots K_n[S]^n}$$

Adair's equation

Four scenarios of relations between binding sites

Identical and independent binding sites

- Assuming binding to each site is independent of the states of other binding sites.
- In steady state, there are **n** possible binding sites available for binding the first substate, only **1** possibility to loose a substrate going from P1 to P_0.

$$0 = -nk_+[P_o][S] + k_-[P_1]$$

- Similarly we get

$$0 = -(n-1)k_+[P_1][S] + 2k_-[P_2]$$

- If intrinsic association constant K is defined as: $K \equiv \frac{k_+}{k_-}$ we have

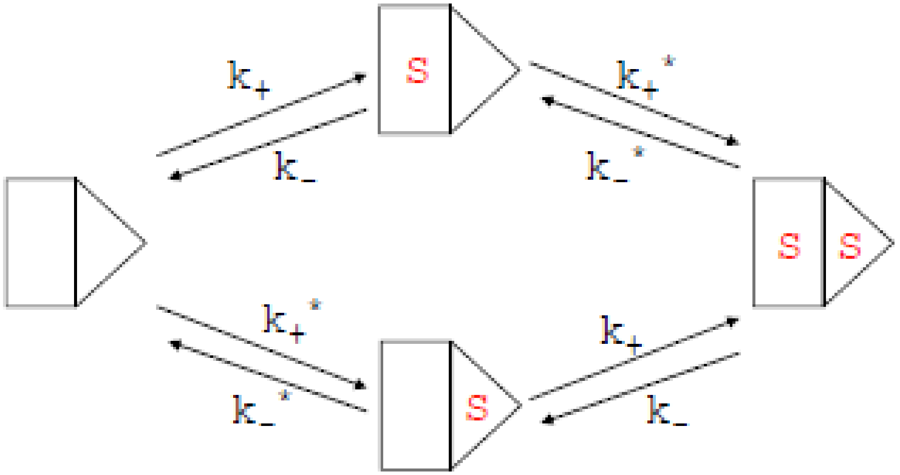
$$K_j = \frac{(n-j+1)K}{j}$$ And the final result is: $$r = \frac{nK[S]}{1+K[S]}$$

Non-identical and independent binding sites

- Each binding site family (n_j) has its own association constant K_j.
- At lower concentration the high affinity binding sites will be occupied, the lower affinity site will be occupied at larger [S]

$$r = \frac{n_1 K_1 [S]}{1 + K_1 [S]} + \frac{n_2 K_2 [S]}{1 + K_2 [S]} + \dots + \frac{n_m K_m [S]}{1 + K_m [S]}$$

$$\sum n_j = n$$



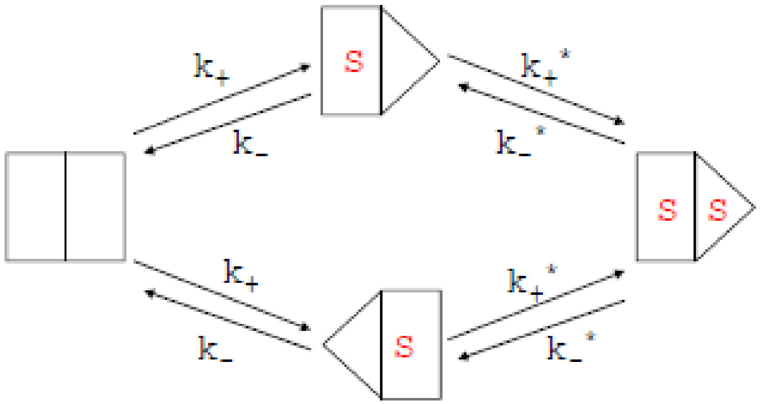
Two identical and interacting binding sites

$$P_0 + S \underset{\mathbf{k_-}}{\overset{\mathbf{k_+}}{\rightleftarrows}} P_1, \qquad P_1 + S \underset{\mathbf{k^*_-}}{\overset{\mathbf{k^*_+}}{\rightleftarrows}} P_2$$

- Two intrinsic association constants $K=k_+/k_-$, $K^*=k^*_+/k^*_-$, we find:

$$K_1 = 2K$$

$$K_2 = \frac{1}{2}K^*$$



- Together with Adair’s equation, the saturation function Y=r/n is:

$$Y = \frac{K[S] + KK^*[S]^2}{1 + 2K[S] + KK^*[S]^2}$$

Two identical and interacting binding sites

- For K=K* we have the Michaelis-Menten like equation:

$$\tilde{Y} = \frac{K[S]}{1+K[S]}$$

- The difference between the two is:

$$Y - \tilde{Y} = \frac{(K^{*} - K)K[S]^{2}}{(1+K[S])\left(1+2K[S]+KK^{*}[S]^{2}\right)}$$

- Positive cooperativity is often defined as $Y - \tilde{Y} > 0$
  - The affinity of binding the second ligand is higher than the first

- Negative cooperativity is often defined as $Y - \tilde{Y} < 0$
  - The affinity of binding the second ligand is lower than the first

Two identical and interacting binding sites

- Further considering the limit for which intermediate states can be neglected, meaning single-bound states are very unlikely, the effective reaction would be:

$$P_o + 2S \leftrightarrow P_2$$

- The saturation function is now:

$$Y = \frac{K[S]^2}{1+K[S]^2}$$

  - Where $K=[P_2]/([P_0][S]^2)$ is the association constant of the reaction. This limit was first considered by Hill. It is called hill function and 2 in this function is called hill coefficient.

Hill coefficient and number of binding sites

Hill coefficient is often used as an estimation of the number of binding sites of an protein.

Just be careful about the assumption of no intermediate states.

Non-identical and interacting binding sites

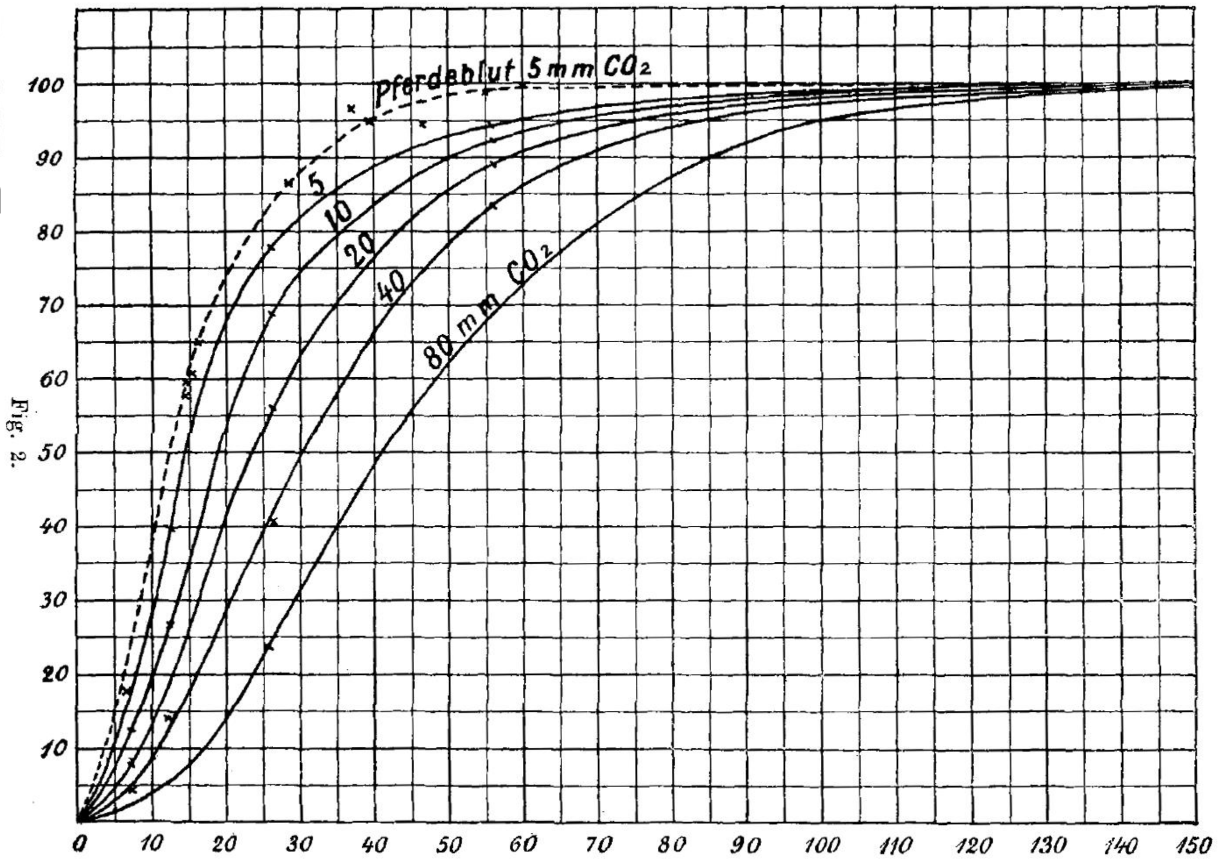
- Beyond the scope of this class
- If possible, experimental determination is better.
- Further reading on enzyme kinetics and coopertivity
  - D. Fell, Understanding the control of metabolism (1997)
  - J.D. Murray, Mathematical Biology (2002)
  - H. Bisswanger, Enzyme kinetics (2002)

Examples of cooperativity binding in biology

Hemoglobin

1. In 1904, Christian Bohr studied hemoglobin binding to oxygen under different condition.

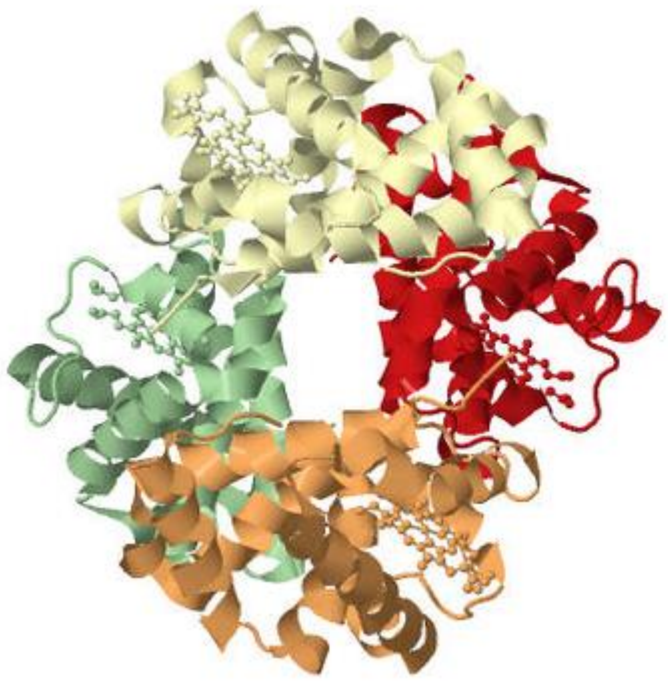
Sigmoidal curve:
The more oxygen is bound to hemoglobin, the easier it is for more oxygen binding until all binding sites are saturated.

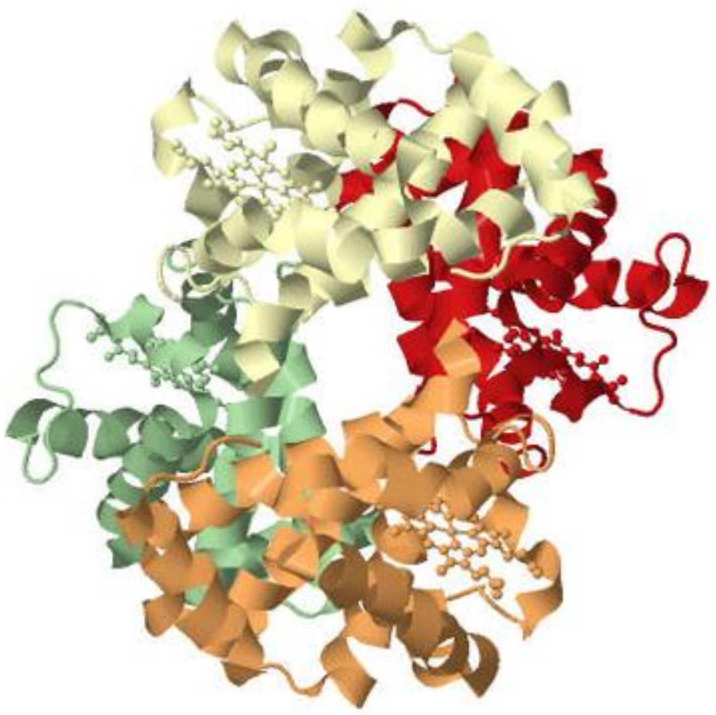


Examples of cooperativity binding in biology

Hemoglobin

2. The molecular evidence for cooperativity: In 1960, Max Perutz solved the tetrameric hemoglobin structure contain four hemes for oxygen binding.

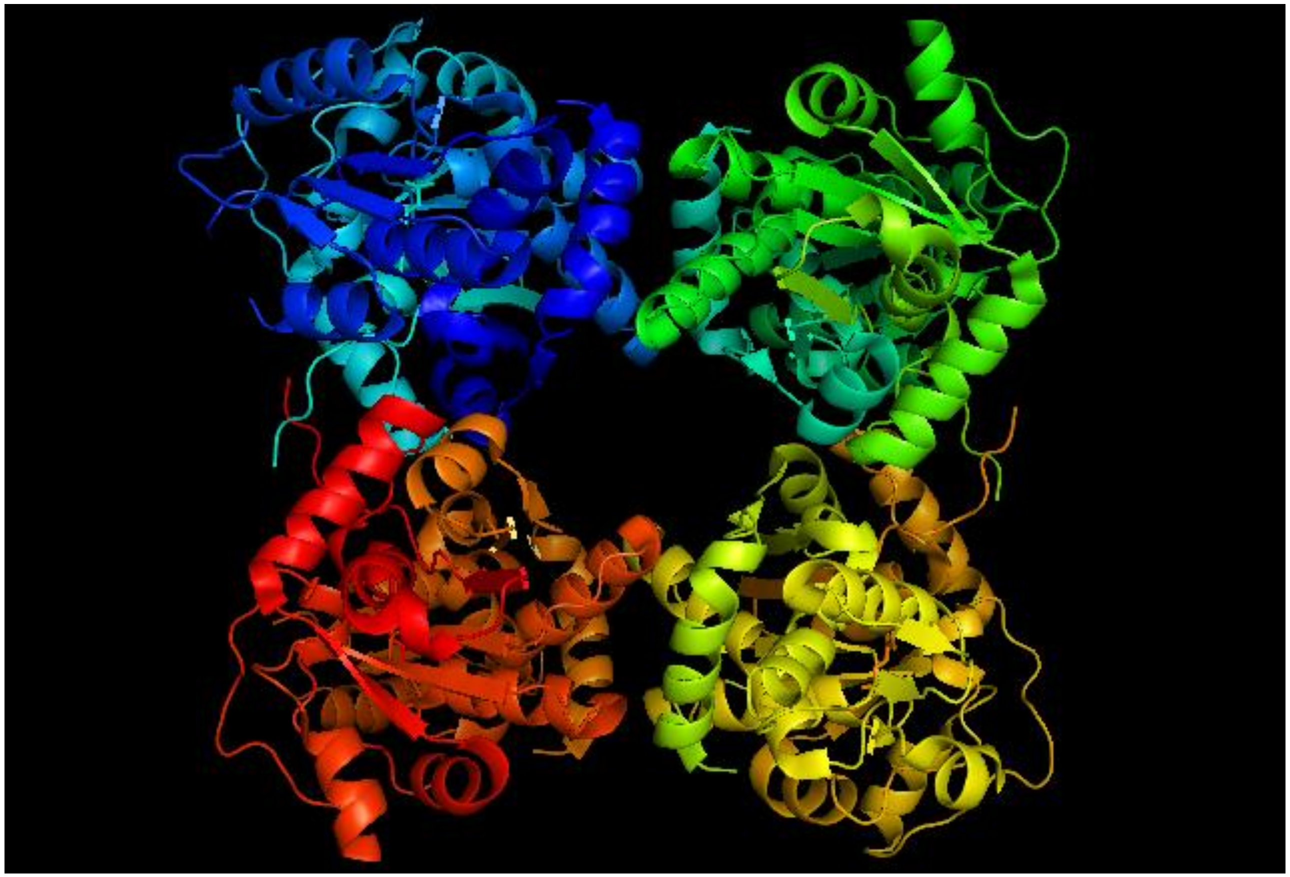

Free


Oxygen-bound

Examples of cooperativity binding in biology

Multimeric enzymes carry several binding sites for the regulator

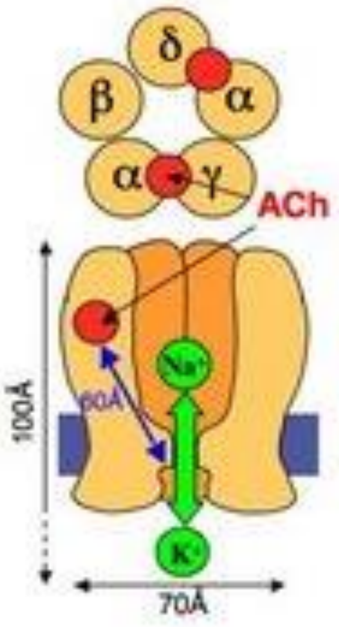
Threonine deaminase

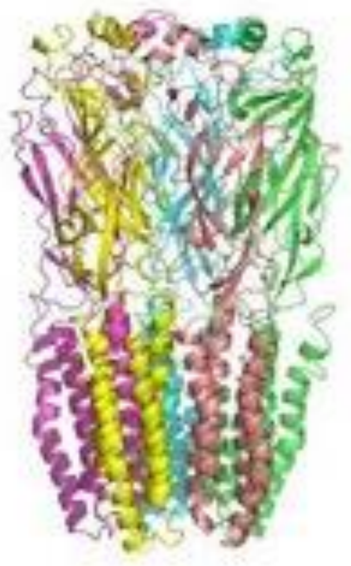


Examples of cooperativity binding in biology

Ion Channels: are formed of several identical monomers, arranged symmetrically in membrane. Several classes of such channel whose opening is regulated by ligands exhibit cooperative binding.

Nicotinic acetylcholine receptors



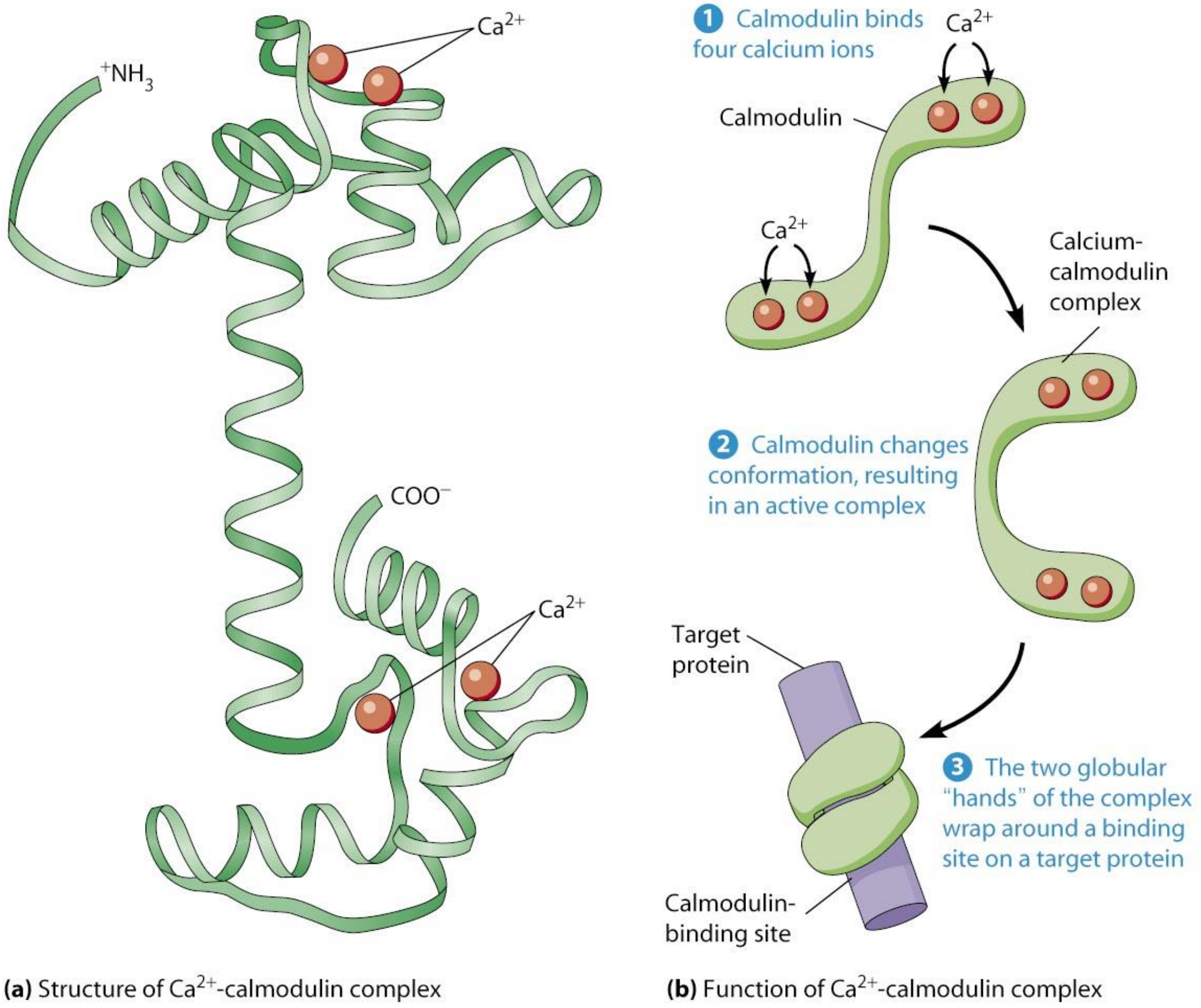


Examples of cooperativity binding in biology

Multi-site molecules:

Calmodulin

Transcriptional factors as well

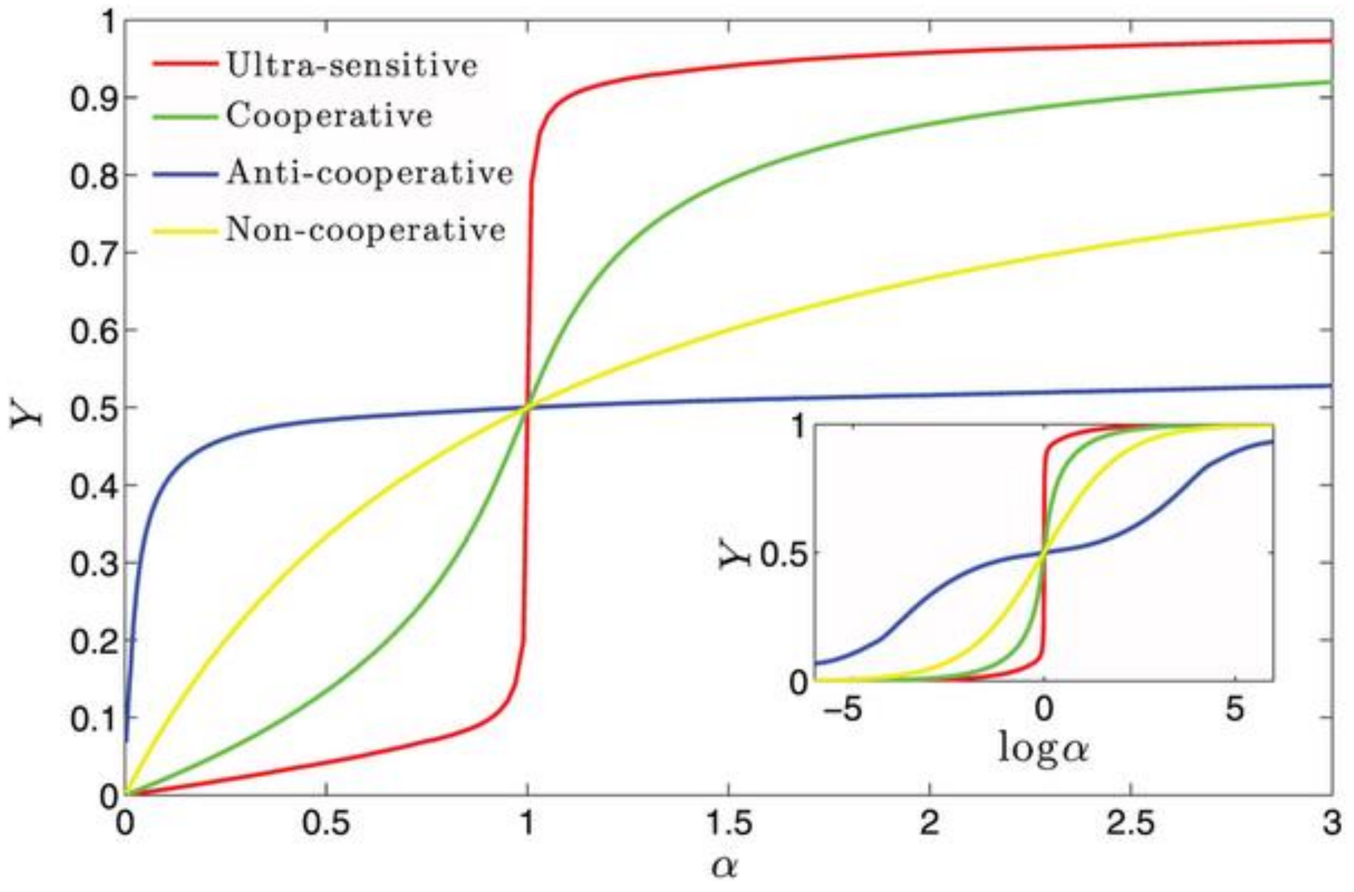


Ca^{2+}
$^+NH_3$
COO^-
Ca^{2+}
1 Calmodulin binds four calcium ions
Ca^{2+}
Calmodulin
Ca^{2+}
Calcium-calmodulin complex
2 Calmodulin changes conformation, resulting in an active complex
Target protein
3 The two globular "hands" of the complex wrap around a binding site on a target protein
Calmodulin-binding site

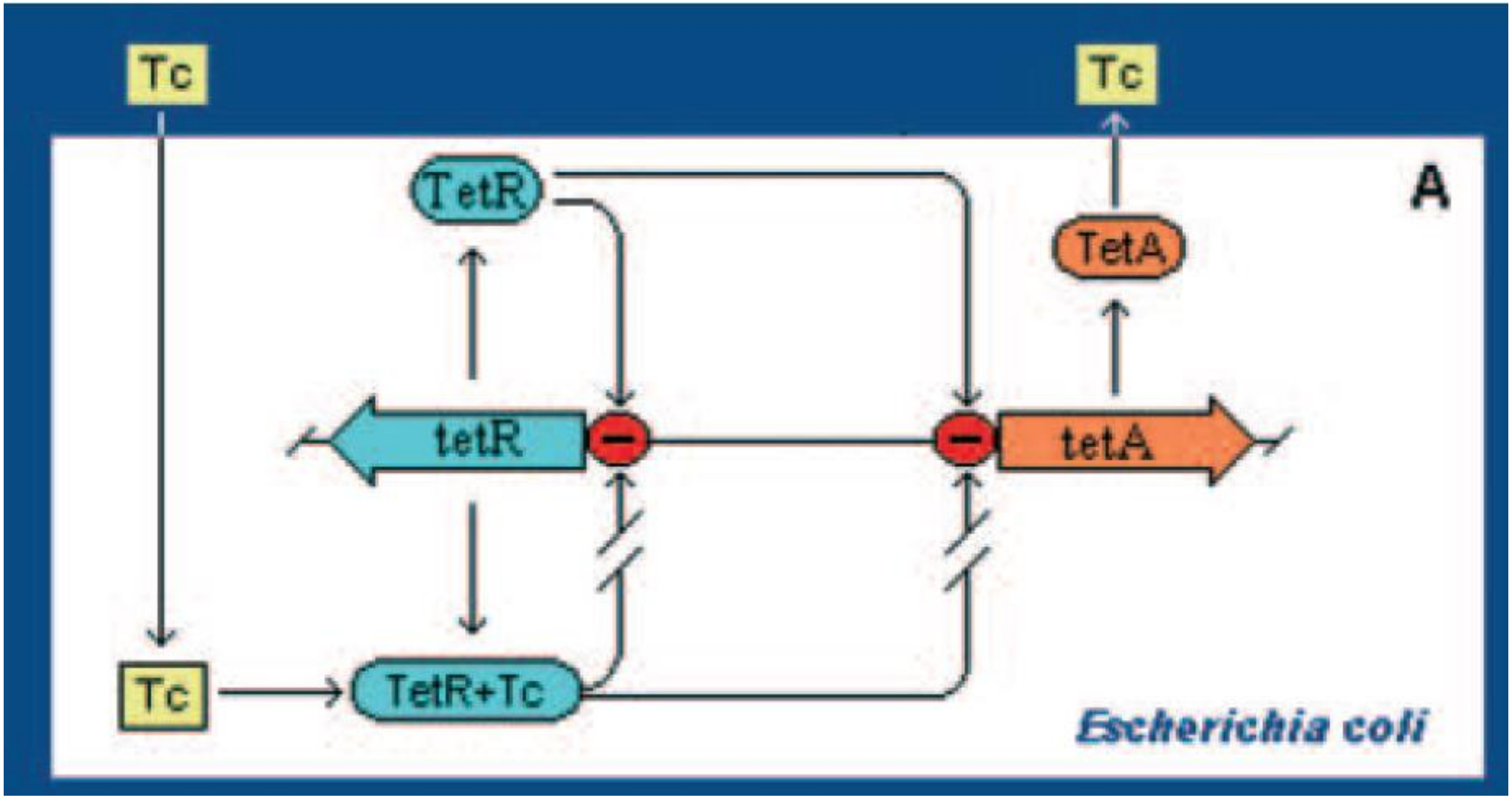
(a) Structure of Ca^{2+}-calmodulin complex **(b)** Function of Ca^{2+}-calmodulin complex

What is cooperativity for (function)?

Sensitivity



Another Example: TetR
Mechanisms for tetracycline resistant



TetR Crystal structure

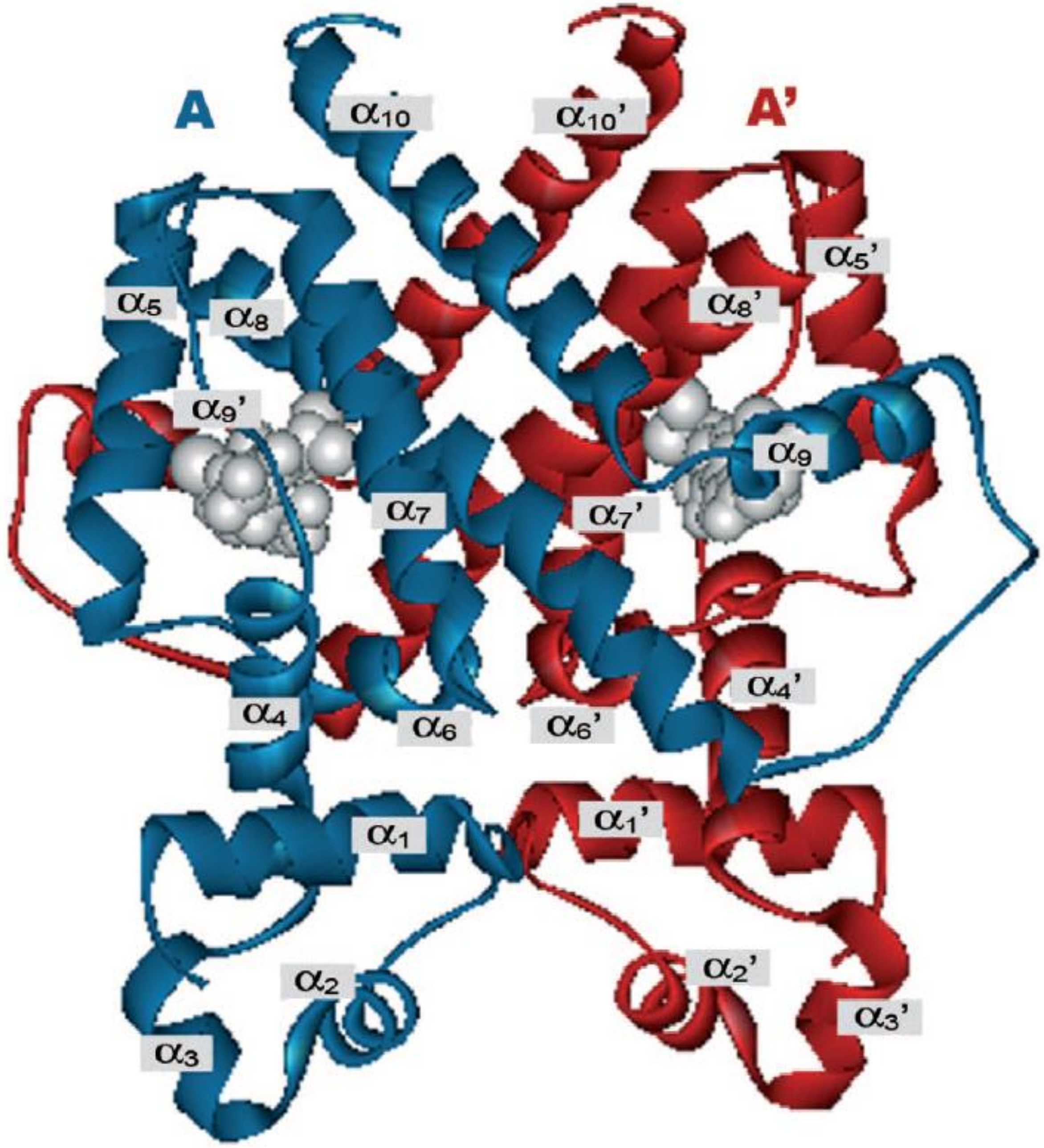


FIG. 2. Ribbon diagram of a TetR homodimer. Monomers are shown in blue or red. Two tetracycline molecules, each bound to a monomer, are shown in grey. α-Helices 2 and 3 in the blue monomer and α2′ and α3′ in the red monomer constitute the shared HTH DNA binding domain. α-Helix 1 and part of helix α-4, together with α-helices 2 and 3, comprise the sequence that best defines the TetR family profile. (Adapted from Hinrichs et al. [150] with permission of the publisher.)

TetR dimer Bind to Palindromic DNA sequence- TetO

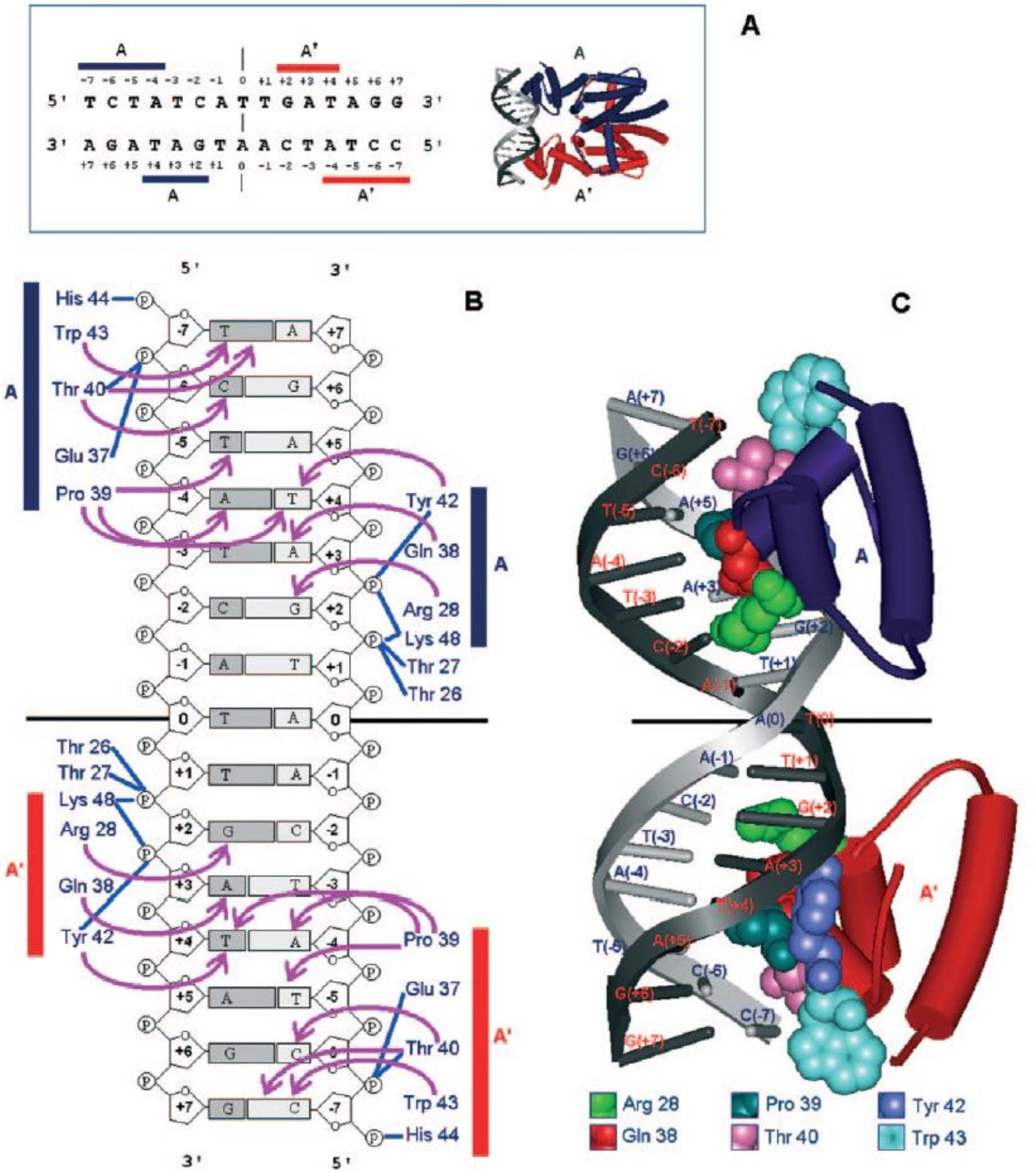


FIG. 3. Binding of TetR to its operator site. A) *tetR* operator and contact regions. The *tetR* operator is a palindromic sequence. Horizontal bars show nucleotides contacted by each monomer of the TetR dimer. B) Interaction of TetR residues with specific nucleotides (arrows) and phosphate backbone (blue lines) in the operator region. The amino acids involved in DNA binding extend from residues 27 to 48. Contacts established with

Cooperative binding of tetracycline to TetR

$$\mathrm{T} + \mathrm{M} \overset{K_M}{\rightleftharpoons} \mathrm{TM}$$

$$\mathrm{TM} + \mathrm{R}_0 \overset{K_1}{\rightleftharpoons} \mathrm{R}_1$$

$$\mathrm{TM} + \mathrm{R}_1 \overset{K_2}{\rightleftharpoons} \mathrm{R}_2 \qquad [1]$$

where T, M and TM represent free tetracycline, Mg^{2+}, and tetracycline complexed by Mg^{2+}, respectively. R_0, R_1, and R_2 are the free repressor dimer, repressor dimer with one TM, and repressor dimer with two TM, respectively. K_M, K_1, and K_2 are the equilibrium association constants of the respective reactions. Since the repressor dimer consists of two chemically identical subunits (14), the two binding sites can be considered intrinsically identical, having an association constant K for tetracycline. The binding of the first tetracycline may modify the binding affinity of the second tetracycline to the repressor to $K \times \alpha$. The macroscopic binding constants K_1 and K_2 are then related to K and α by the equations

$$K_1 = 2 \times K$$

$$K_2 = K \times \alpha/2.$$

Cooperative binding of tetracycline to TetR

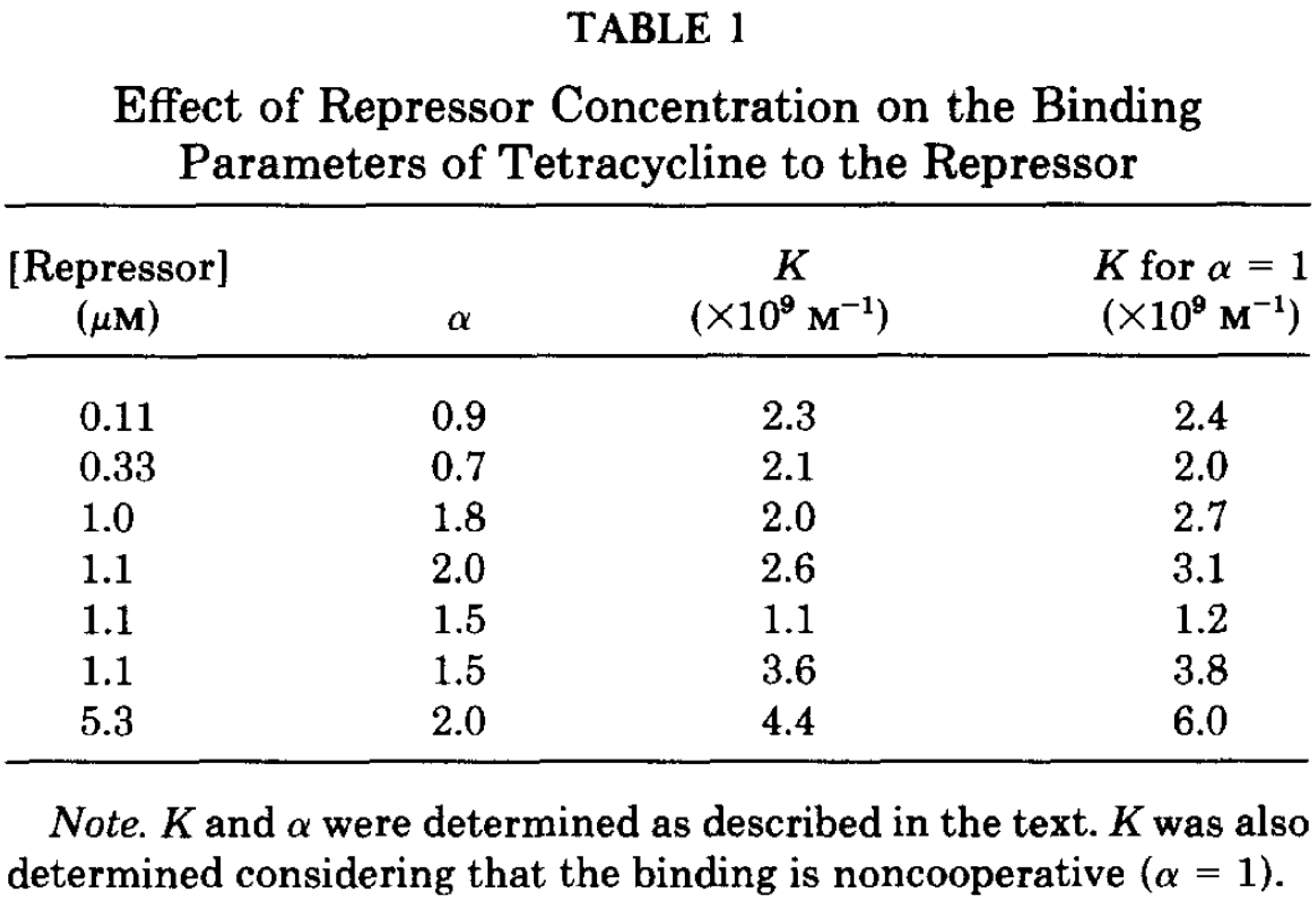


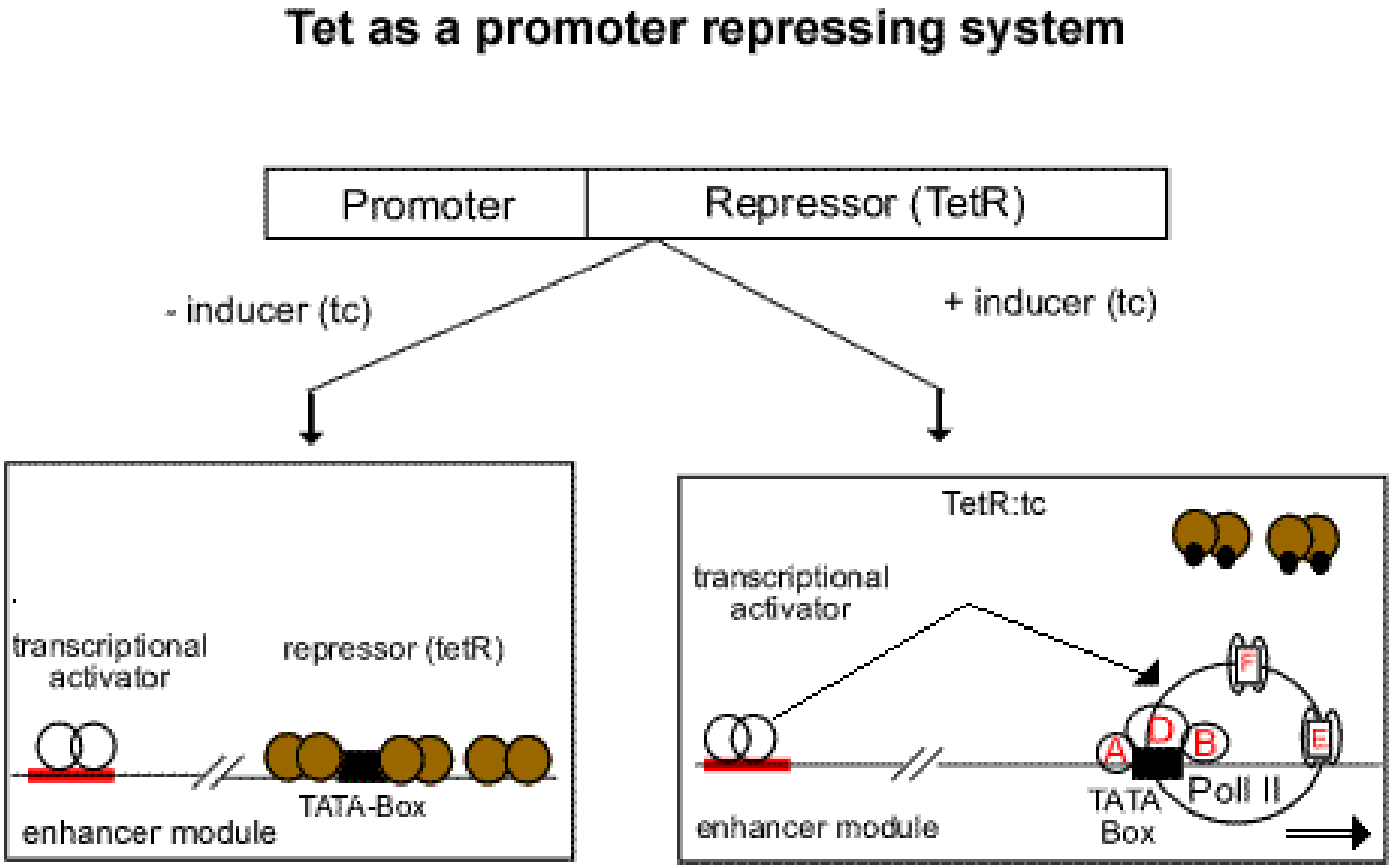
TABLE 1

Effect of Repressor Concentration on the Binding Parameters of Tetracycline to the Repressor

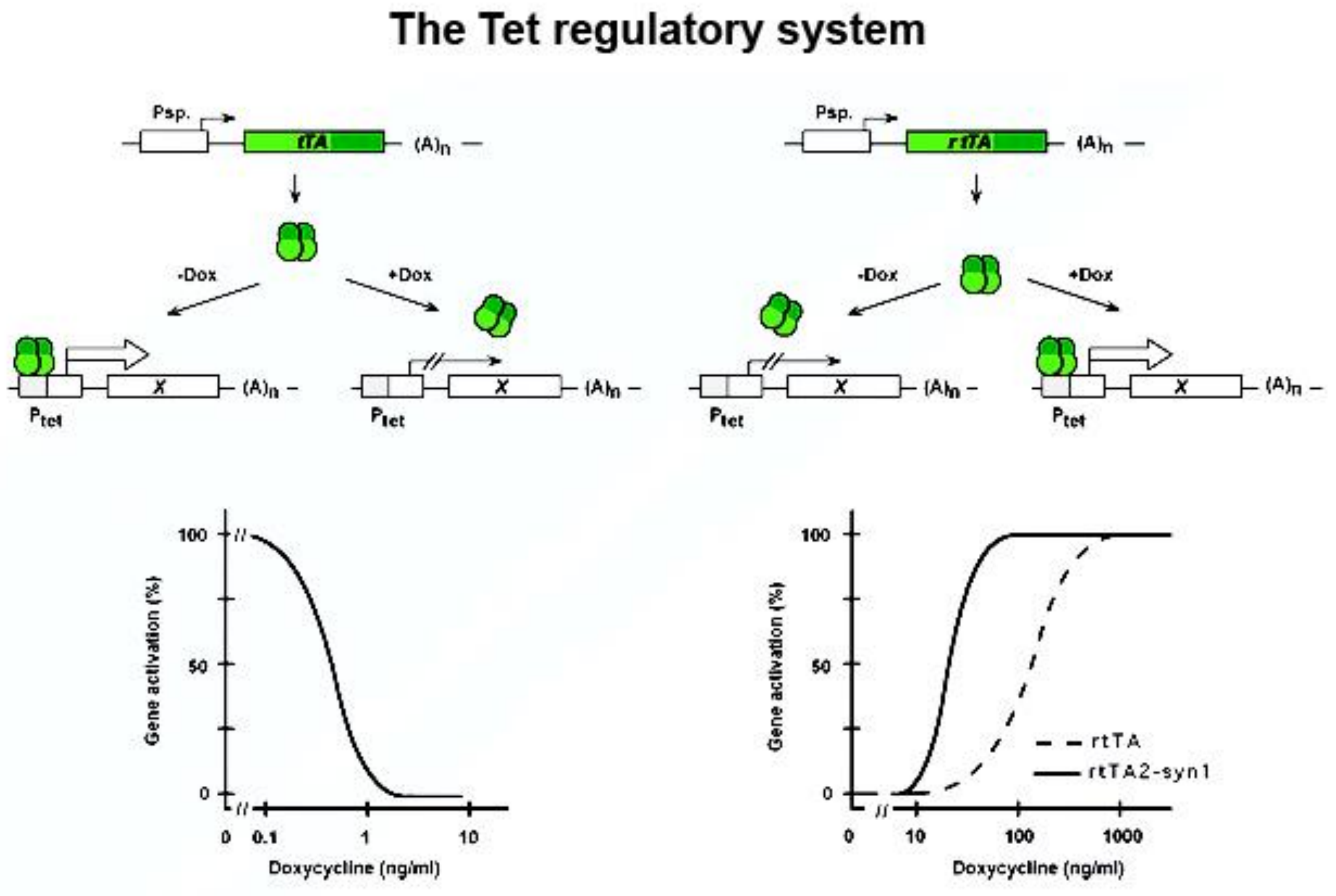
[Repressor] (μM)	α	K ($\times 10^9$ M^{-1})	K for $\alpha = 1$ ($\times 10^9$ M^{-1})
0.11	0.9	2.3	2.4
0.33	0.7	2.1	2.0
1.0	1.8	2.0	2.7
1.1	2.0	2.6	3.1
1.1	1.5	1.1	1.2
1.1	1.5	3.6	3.8
5.3	2.0	4.4	6.0

Note. K and α were determined as described in the text. K was also determined considering that the binding is noncooperative ($\alpha = 1$).

Induced expression system with WT tetR



Induced expression system with engineered tetR


The Tet regulatory system
Psp.
tTA
(A)n
-Dox
+Dox
Ptet
X
rtTA
Gene activation (%)
100
50
0
0
0.1
1
10
Doxycycline (ng/ml)
10
100
1000
rtTA
rtTA2-syn1