Table of Contents


```
% Robert Heeter
% BIOE 391 Numerical Methods
% EXAM 1 MATLAB SCRIPT

clc, clf, clear, close all
```

PROBLEM 1, PART A

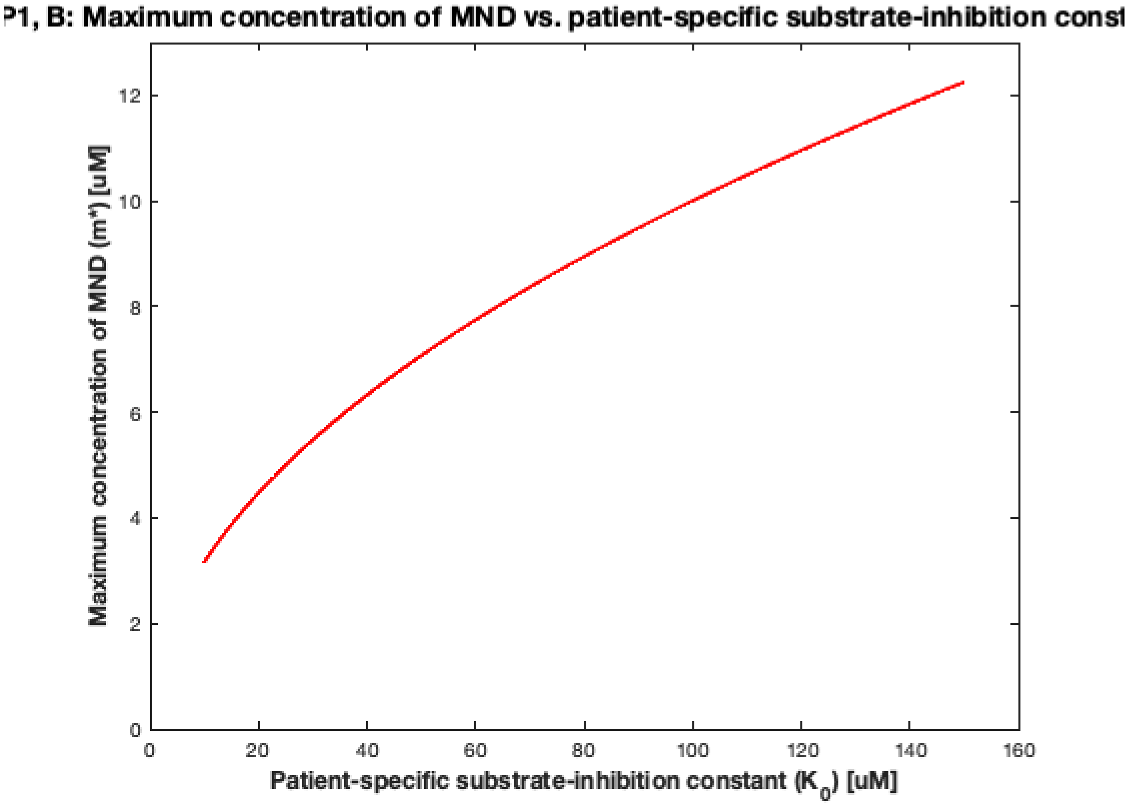
```
disp('PROBLEM 1');

% Constants and equation
V_max = 1.25; % maximum degradation rate (1/hr)
K = 1; % Michaelis-Menten constant characterizing saturation of liver
 enzymes (uM)
Vm = @(m,K_0) (V_max.*m)./(K+m+(m.^2./K_0)); % kinetic equation
 as function of m (concentration of MND) and K_0 (patient-specific
 substrate-inhibition constant)

% Graphical method to find initial point for fminsearch
figure
K_0_temp = 10;
fplot(@(m) Vm(m,K_0_temp),[-20, 20],'-b','LineWidth',2);
xlabel('Concentration of MND (m)
 [uM]','FontSize',12,'FontWeight','bold');
ylabel('Clearance rate (V) [1/hr]','FontSize',12,'FontWeight','bold');
title('P1, A: Graphical method for fminsearch initial point (V vs.
 m)','FontSize',14,'FontWeight','bold');
ax = gca;
ax.XAxisLocation = 'origin';
ax.YAxisLocation = 'origin';

guess1 = 0; % graphical method shows approximate maximum near m ~ 0

% Example of fminsearch function, using initial point m ~ 0
K_0 = 10; % sample K_0 value to test vm_fminsearch function
[V_star,m_star] = vm_fminsearch(Vm,guess1,K_0); % vm_fminsearch
 function written and end of document
```

```
% Display results
disp('P1,A: Fminsearch method example output:');
fprintf('K_0 [uM] = %d\nmaximum (m_star, V_star) [uM, 1/hr] = (%f,
 %f)\n\n',K_0,m_star,V_star);
```

PROBLEM 1

PROBLEM 1, PART B

```
% Determine points
guess1 = 0; % graphical method in part A shows approximate maximum
 near m ~ 0
K_0_int = (10:0.1:150)'; % interval of K_0 values
m_star_int = zeros(size(K_0_int)); % preallocate
for i = 1:length(K_0_int)
    [~,m_star_int(i)] = vm_fminsearch(Vm,guess1,K_0_int(i)); % find
 value of m_star at each K_0 value
end

% Make figure
figure
plot(K_0_int,m_star_int,'-r','LineWidth',2);
xlabel('Patient-specific substrate-inhibition constant (K_0)
 [uM]','FontSize',12,'FontWeight','bold');
ylabel('Maximum concentration of MND (m*)
 [uM]','FontSize',12,'FontWeight','bold');
title('P1, B: Maximum concentration of MND vs. patient-specific
 substrate-inhibition constant','FontSize',14,'FontWeight','bold');
axis([0 160 0 13]);
```

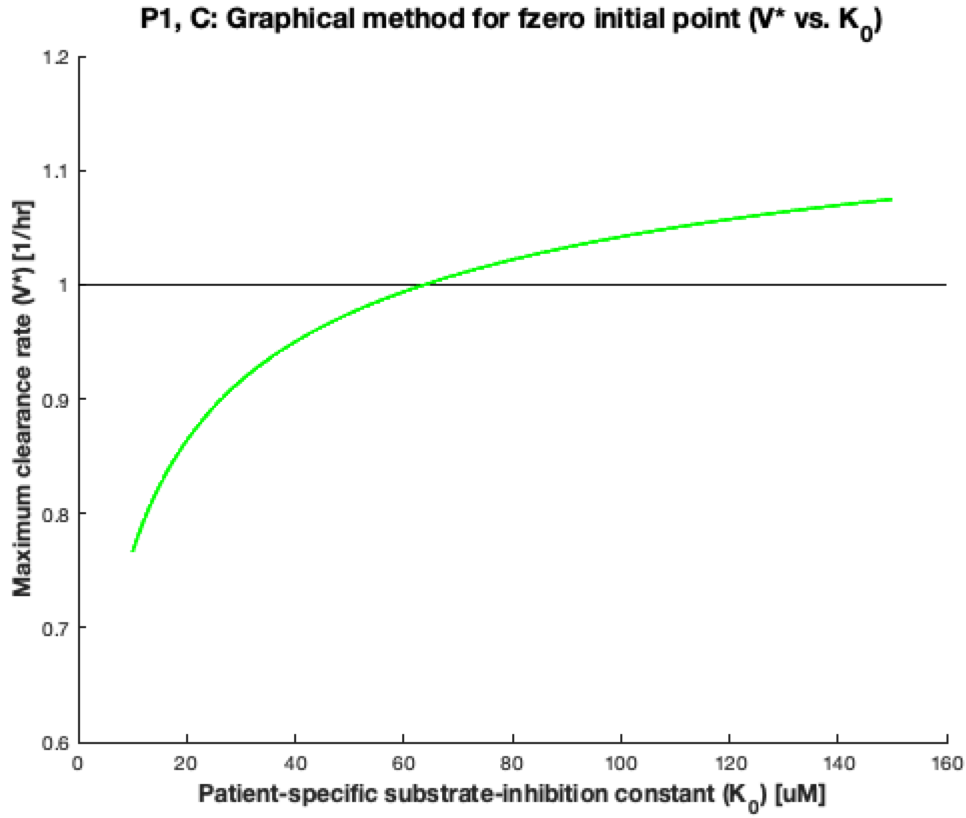


PROBLEM 1, PART C

```
% Graphical method to find initial point for fzero
V_star_int = Vm(m_star_int,K_0_int); % use interval of m_star values
 from part B

figure
hold on
fplot(1,'-k','LineWidth',1);
plot(K_0_int,V_star_int,'-g','LineWidth',2);
xlabel('Patient-specific substrate-inhibition constant (K_0)
 [uM]','FontSize',12,'FontWeight','bold');
ylabel('Maximum clearance rate (V*) [1/
hr]','FontSize',12,'FontWeight','bold');
title('P1, C: Graphical method for fzero initial point (V* vs.
 K_0)','FontSize',14,'FontWeight','bold');
axis([0 160 0.6 1.2]);
hold off

guess2 = 65; % graphical method shows V* ~ 1 near K_0 = 65

% Use fzero to find K_0 that gives V* = 1.0/hr
[K_0_crit,V_star_rel] = fzero(@(K_0) 1-
(vm_fminsearch(Vm,guess1,K_0)),guess2); % minimize (1-maximum of Vm)
 depending on K_0
```

```
% Display results
disp('P1,C: Fzero method output:')
fprintf('K_0 [uM] = %f\nV_star [1/hr] = %f\n\n',K_0_crit,V_star_rel
+1);
```

```
P1,C: Fzero method output:
K_0 [uM] = 64.000000
V_star [1/hr] = 1.000000
```

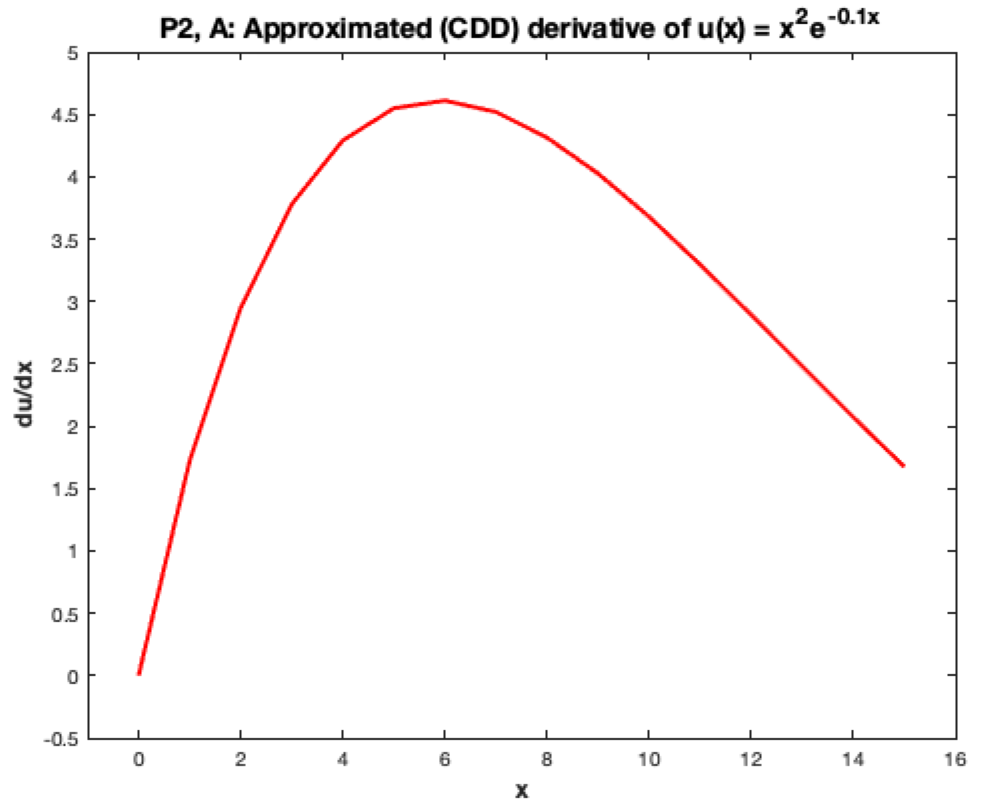


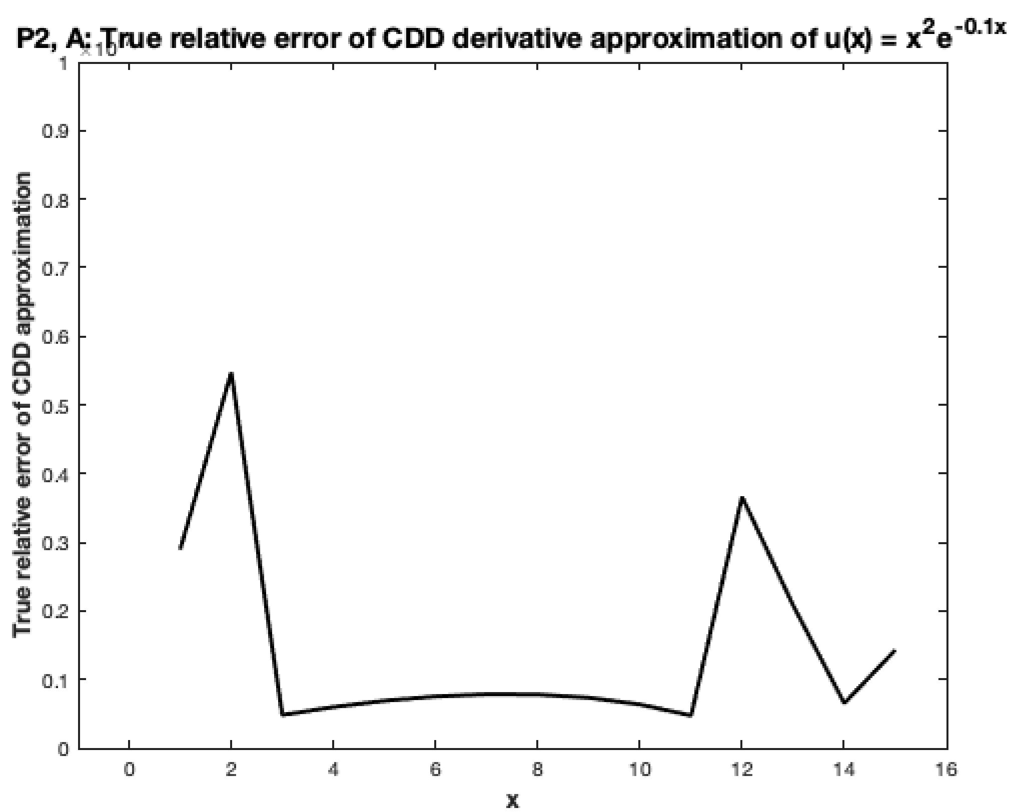
PROBLEM 2, PART A

```
disp('PROBLEM 2');

% Example shear strain equation and analytical derivative
ux = @(x) (x.^2).*exp(-0.1.*x);
dudx = @(x) (2.*x.*exp(-0.1.*x)) - (0.1.*(x.^2).*exp(-0.1.*x));

% Determine points
x_int = (0:1:15)';
dudx_cdd = zeros(size(x_int));
for i = 1:length(x_int)
    dudx_cdd(i) = shearstrain(ux,x_int(i));
end
dudx_exact = dudx(x_int);
```

```
er = abs(dudx_exact - dudx_cdd)./dudx_exact;

% Display results
disp('P2,A: Comparing CDD approximation and analytical derivative:') %
 display results
fprintf(' x:         du/dx (approx):   du/dx (exact):   True rel.
 error: \n');
fprintf(' %4.1f     %7.4f           %7.4f           %g\n',
[x_int,dudx_cdd,dudx_exact,er]');
disp(' ');

% Plot results
figure
plot(x_int,dudx_cdd,'-r','LineWidth',2);
xlabel('x','FontSize',12,'FontWeight','bold');
ylabel('du/dx','FontSize',12,'FontWeight','bold');
title('P2, A: Approximated (CDD) derivative of u(x) =
 x^2e^{-0.1x}','FontSize',14,'FontWeight','bold');
axis([-1 16 -0.5 5]);

figure
plot(x_int,er,'-k','LineWidth',2);
xlabel('x','FontSize',12,'FontWeight','bold');
ylabel('True relative error of CDD
 approximation','FontSize',12,'FontWeight','bold');
title('P2, A: True relative error of CDD derivative approximation of
 u(x) = x^2e^{-0.1x}','FontSize',14,'FontWeight','bold');
axis([-1 16 0 1e-7]);
```

```
PROBLEM 2
P2,A: Comparing CDD approximation and analytical derivative:
 x:         du/dx (approx):   du/dx (exact):   True rel. error:
  0.0       0.0000            0.0000           NaN
  1.0       1.7192            1.7192           2.8965e-08
  2.0       2.9474            2.9474           5.47057e-08
  3.0       3.7782            3.7782           4.8133e-09
  4.0       4.2900            4.2900           5.97614e-09
  5.0       4.5490            4.5490           6.88775e-09
  6.0       4.6100            4.6100           7.52041e-09
  7.0       4.5189            4.5189           7.83993e-09
  8.0       4.3136            4.3136           7.79906e-09
  9.0       4.0250            4.0250           7.33533e-09
 10.0       3.6788            3.6788           6.35801e-09
 11.0       3.2954            3.2954           4.73955e-09
 12.0       2.8915            2.8915           3.66213e-08
 13.0       2.4800            2.4800           2.0781e-08
 14.0       2.0714            2.0714           6.5271e-09
 15.0       1.6735            1.6735           1.43052e-08
```

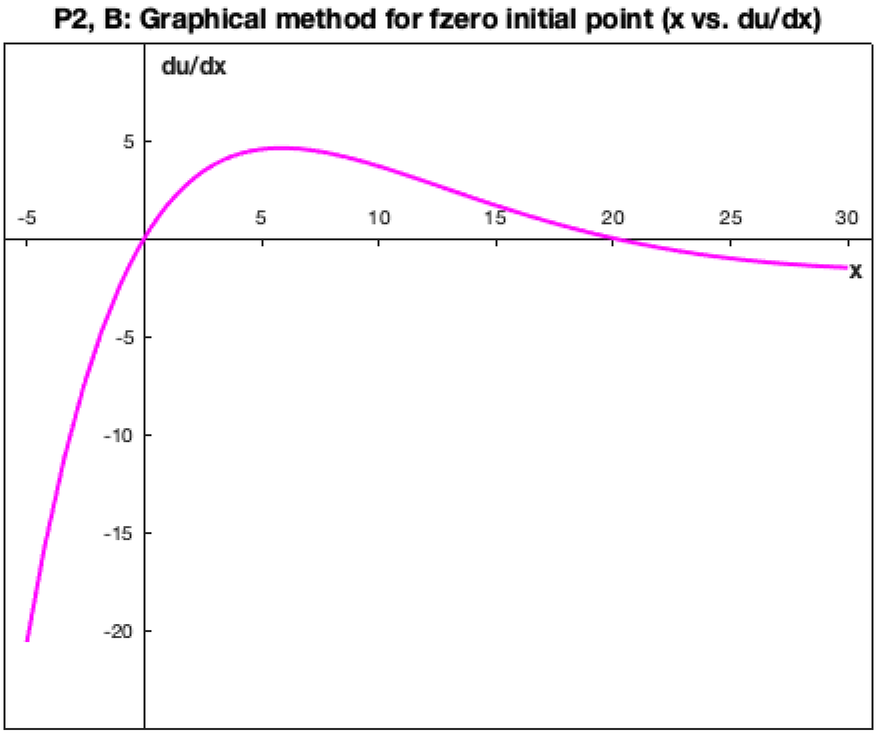

P2, A: Approximated (CDD) derivative of u(x) = x2e-0.1x
du/dx
x
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
0
2
4
6
8
10
12
14
16


P2, A: True relative error of CDD derivative approximation of u(x) = x2e-0.1x
True relative error of CDD approximation
x
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
2
4
6
8
10
12
14
16

PROBLEM 2, PART B

```
% Graphical method to find initial point for fzero
figure
fplot(dudx,[-5,30],'-m','LineWidth',2);
xlabel('x','FontSize',12,'FontWeight','bold');
ylabel('du/dx','FontSize',12,'FontWeight','bold');
title('P2, B: Graphical method for fzero initial point (x vs. du/
dx)','FontSize',14,'FontWeight','bold');
ax = gca;
ax.XAxisLocation = 'origin';
ax.YAxisLocation = 'origin';
axis([-6 31 -25 10]);

guess1 = 20; % graphical method shows du/dx ~ 0 near x = 20

% Use fzero to find du/dx = 0
[x_root,dudx_root] = fzero((@(x) shearstrain(ux,x)),guess1); %
 minimize shearstrain function (CDD derivative approximation)

% Display results
disp('P2,B: Fzero method output:');
fprintf('root of dudx = (%f, %f)\n\n',x_root,dudx_root);
```

```
P2,B: Fzero method output:
root of dudx = (20.000000, 0.000000)
```



PROBLEM 2, PART C

```
% Use fminbnd to find maximum of u(x)
guess_lower = 10; % upper and lower guesses given maximum occurs
 around root (x = 20);
guess_upper = 30;

[x_max,ux_max] = fminbnd(@(x) -1*ux(x),guess_lower,guess_upper); % use
 fminbnd for negative u(x) for maximum
ux_max = -1*ux_max; % ux_max is -1*fminbnd output

% Display results
disp('P2,C: Fminbnd method output:');
fprintf('max of u(x) = (%f, %f)\n\n',x_max,ux_max);
```

```
P2,C: Fminbnd method output:
max of u(x) = (20.000014, 54.134113)
```

Additional Functions

```
function [V_star, m_star] = vm_fminsearch(Vm,guess,K_0)
% ABOUT: Implements fminsearch to find maximum of function Vm near an
% initial value.
```

```
% INPUTS: Vm = function; guess = initial point; K_0 = K_0 parameter
 for Vm
% OUTPUTS: V_star = Vm-value at maximum; m_star = m-value at maximum

[m_star,V_temp] = fminsearch(@(m) -1*Vm(m,K_0),guess); % use
 fminsearch for negative function for maximum
V_star = -1*V_temp; % V_star is -1*fminbnd output

end

function [dudx] = shearstrain(ux,x0,h,ea)
% ABOUT: Implements centered divided-difference approximation for
% derivative of u(x) at x0 for incrementally smaller step sizes (h)
% INPUTS: ux = function; x0 = point for derivative; h = step size; ea
 =
% approximate relative error threshold
% OUTPUTS: dudx = derivative approximation at x0

% Check inputs
if nargin < 2 || isempty(ux) || isempty(x0)
    error('At least 2 input arguments required.')
end

if nargin < 3 || isempty(h)
    if x0 ~= 0
        h = abs(x0)/10;
    else
        h = 1;
    end
end

if nargin < 4 || isempty(ea)
    ea = 0.0001;
end

% Iterate CDD until iteration or error thresholds are reached
dudx_old = (ux(x0+h) - ux(x0-h))/(2*h); % centered divided difference
 formula
maxit = 50; % maximum number of iterations
iter = 0;
er = 100;

while (1)
    h = h/2;
    iter = iter+1;
    dudx = (ux(x0+h) - ux(x0-h))/(2*h); % centered divided difference
 formula
    if dudx ~= 0
        er = abs((dudx-dudx_old)/dudx)*100;
    end
    if er < ea || iter >= maxit
        break
    end
    dudx_old = dudx;
```

```
    h = h/2;
end

end
```

P1,A: Fminsearch method example output:
K_0 [uM] = 10
maximum (m_star, V_star) [uM, 1/hr] = (3.162250, 0.765718)

Published with MATLAB® R2021a