《机械工程中的数值分析技术》

作业

学　　生：易弘睿

学　　号：20186103

专业班级：机械一班

作业编号：2021072009

重庆大学-辛辛那提大学联合学院

二〇二一年七月

Catalog

Chap21 Numerical Differentiation

1.1 Question 21.11

21.11 The following data were collected for the distance traveled versus time for a rocket:

t, s	0	25	50	75	100	125
y, km	0	32	58	78	92	100

Use numerical differentiation to estimate the rocket's velocity and acceleration at each time.

The Matlab code is below:

```
clc;clear all;
t = [0, 25, 50, 75, 100, 125];
y = [0, 32, 58, 78, 92, 100];
[v, a] = diffeq(t, y);
fprintf("Velocity(m/s)\tAcceleration(m/s^2)\n")
fprintf("%0.4f\t\t\t%0.4f\n", [v' a']')
function [dydx, d2ydx2] = diffeq(x, y)
    n = length(x);
    if length(y) ~= n
        fprintf("Î¬¶È²»Ò»ÖÂ\n")
        return
    end
    dx = diff(x);
    h = dx(1);
    dy = diff(y);
    dydx = zeros(1, n);
    d2ydx2 = zeros(1, n);
    for i=1:n
        if i == n
            dydx(i) = (-3*dy(i-1)+dy(i-2))/(dx(i-1)+dx(i-
2));
            d2ydx2(i) = (2*dy(i-1)-3*dy(i-2)+dy(i-
3))/(dx(i-1)*dx(i-2));
        elseif i == 1
            dydx(i) = (3*dy(i)-dy(i+1))/(dx(i)+dx(i+1));
```

```
            d2ydx2(i) = (-2*dy(i)+3*dy(i+1)-
dy(i+2))/(dx(i)*dx(i+1));
        else
            dydx(i) = (dy(i)+dy(i-1))/(dx(i-1)+dx(i));
            d2ydx2(i) = (dy(i)-dy(i-1))/(dx(i-1)*dx(i));
        end
    end
end
```

The output is below:

```
Velocity(m/s)  Acceleration(m/s^2)
1.4000         -0.0096
1.1600         -0.0096
0.9200         -0.0096
0.6800         -0.0096
0.4400         -0.0096
-0.2000        -0.0096
```

1.2 Question 21.31

21.31 The pressure gradient for laminar flow through a constant radius tube is given by

$$\frac{dp}{dx} = -\frac{8\mu Q}{\pi r^4}$$

where p = pressure (N/m^2), x = distance along the tube's centerline (m), μ = dynamic viscosity ($N \cdot s/m^2$), Q = flow (m^3/s) and r = radius (m).

(a) Determine the pressure drop for a 10-cm length tube for a viscous liquid ($\mu = 0.005\ N \cdot s/m^2$, density $= \rho = 1 \times 10^3\ kg/m^3$) with a flow of $10 \times 10^{-6}\ m^3/s$ and the following varying radii along its length:

***x*, cm**	0	2	4	5	6	7	10
***r*, mm**	2	1.35	1.34	1.6	1.58	1.42	2

(b) Compare your result with the pressure drop that would have occurred if the tube had a constant radius equal to the average radius.

(c) Determine the average Reynolds number for the tube to verify that flow is truly laminar ($Re = \rho v D/\mu < 2100$ where v = velocity).

The Matlab code is below:

```
clear; clc; close all;
x = [0 2 4 5 6 7 10] * 10^-2;
r = [2 1.35 1.34 1.6 1.58 1.42 2] * 10^-3;
u = 0.005;
dens = 1000;
Q = 10*10^-6;
dp_dx = @(r) (-8 * Q * u) ./ (pi * r.^4);
dp_drop = dp_dx(r);
p_drop = 0;
```

```
for i = 1:length(x)-1
    p_drop = p_drop + (dp_drop(i)+dp_drop(i+1))/2 *
(x(i+1)-x(i));
end
r_ave = mean(r) * ones(1, 7);
dp_ave = dp_dx(r_ave);
p_ave = 0;
for i = 1:length(x)-1
    p_ave = p_ave + (dp_ave(i)+dp_ave(i+1))/2 * (x(i+1)-
x(i));
end
A = pi * mean(r)^2;
v = Q / A;
Re = dens * v * mean(r) * 2 / u;
fprintf('(a)The pressure drop for a viscous liquid with a
flow varying radii is %f Pa\n',abs(p_drop))
fprintf('(b)The pressure drop of average radius is %f
Pa\n',abs(p_ave))
fprintf('(c) The average Reynold number is %f.',Re)
if Re < 2100
    disp(' And the flow is truely laminar')
else
    disp(' And the flow is not truely laminar')
end
```

The output is below:

```
(a)The pressure drop for a viscous liquid with a flow
varying radii is 2582.856276 Pa
(b)The pressure drop of average radius is 1881.596563 Pa
(c) The average Reynold number is 789.431073. And the flow
is truely laminar
```

1.3 Question 21.35

21.35 The sediment oxygen demand [SOD in units of $g/(m^2 \cdot d)$] is an important parameter in determining the dissolved oxygen content of a natural water. It is measured by placing a sediment core in a cylindrical container (Fig. P21.35). After carefully introducing a layer of distilled, oxygenated water above the sediments, the container is covered to prevent gas transfer. A stirrer is used to mix the water gently, and an oxygen probe tracks how the water's oxygen concentration decreases over time. The SOD can then be computed as

$$\mathrm{SOD} = -H\frac{do}{dt}$$

where $H =$ the depth of water (m), $o =$ oxygen concentration (g/m^3), and $t =$ time (d).

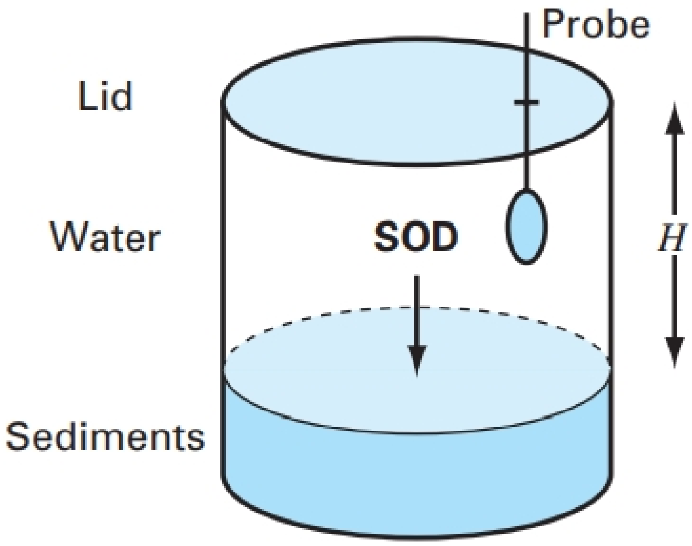


FIGURE P21.35

Based on the following data and $H = 0.1$ m, use numerical differentiation to generate plots of **(a)** SOD versus time and **(b)** SOD versus oxygen concentration:

***t*, d**	0	0.125	0.25	0.375	0.5	0.625	0.75
***o*, mg/L**	10	7.11	4.59	2.57	1.15	0.33	0.03

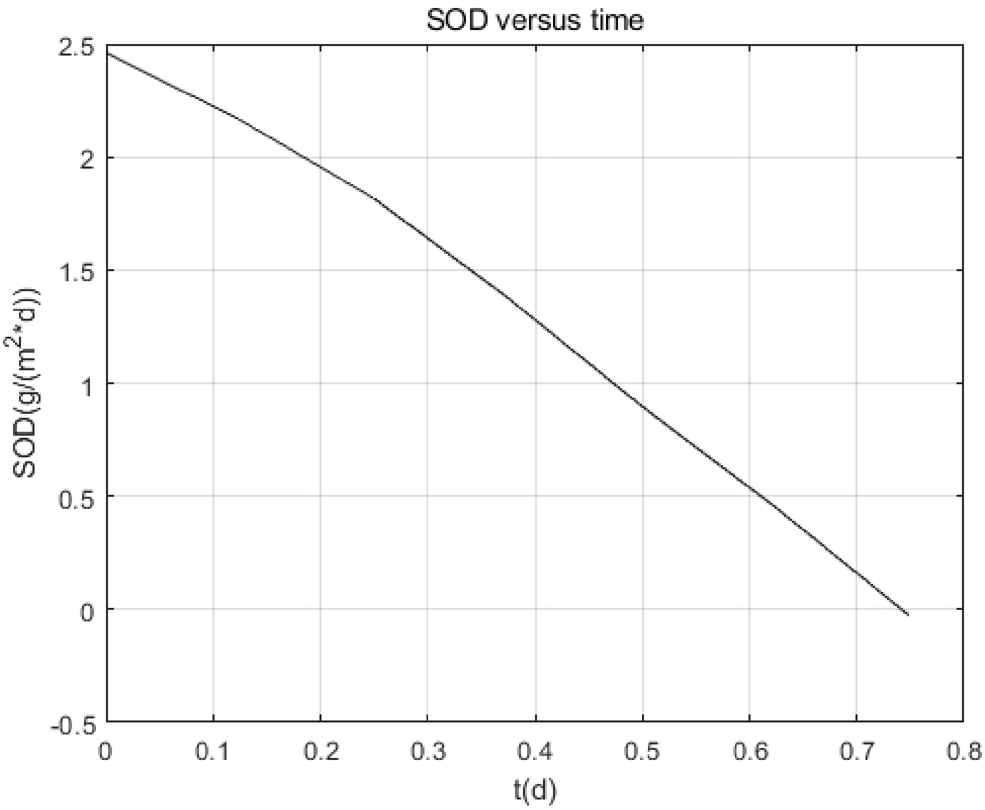
The Matlab code is below:

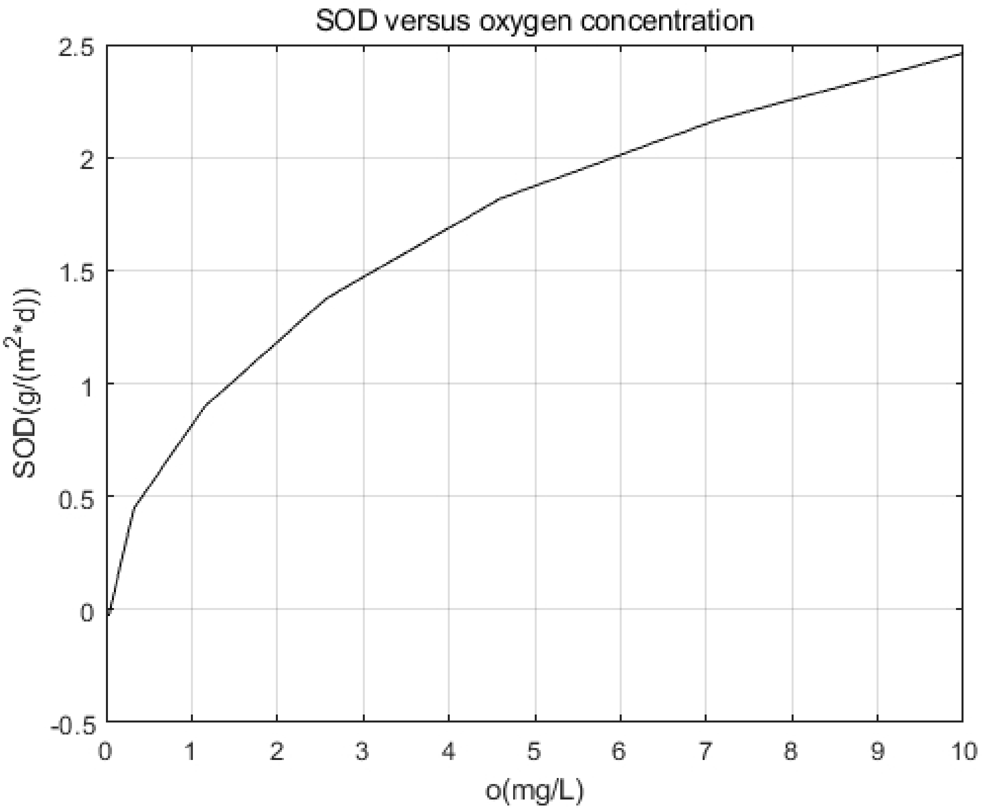
```
clc;clear all;
t = [0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75];
o = [10, 7.11, 4.59, 2.57, 1.15, 0.33, 0.03];
H = 0.1;
dodt = diffeq(t, o);
SOD = -H*dodt;
figure(1)
plot(t, SOD, "-k")
grid on
xlabel("t(d)")
ylabel("SOD(g/(m^2*d))")
title("SOD versus time")
```

```
figure(2)
plot(o, SOD, "-k")
grid on
xlabel("o(mg/L)")
ylabel("SOD(g/(m^2*d))")
title("SOD versus oxygen concentration")

function dydx = diffeq(x, y)
    n = length(x);
    if length(y) ~= n
        fprintf("ÊäÈë²ÎÊýÎ¬¶È²»Ò»ÖÂ\n")
        return
    end
    dx = diff(x);
    dy = diff(y);
    dydx = zeros(1, n);
    for i=1:n
        if i == 1
            dydx(i) = (3*dy(i)-dy(i+1))/(dx(i)+dx(i+1));
        elseif i == n
            dydx(i) = (-3*dy(i-1)+dy(i-2))/(dx(i-1)+dx(i-
2));
        else
            dydx(i) = (dy(i)+dy(i-1))/(dx(i-1)+dx(i));
        end
    end
end
```

The output is below:


SOD versus time
SOD(g/(m^2*d))
t(d)
2.5
2
1.5
1
0.5
0
-0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8


SOD versus oxygen concentration
SOD(g/(m^2*d))
o(mg/L)
2.5
2
1.5
1
0.5
0
-0.5
0
1
2
3
4
5
6
7
8
9
10