TOM BERTALAN

CBE504, Homework 1, due Nov. 12, in class.

Problem 1: Consider the case when both CO and O_2 adsorb on the catalyst surface reversibly. Furthermore assume that reaction between CO and O is slow compared to the adsorption steps, which can be considered in equilibrium. Derive the expression for the rate of CO oxidation under these assumptions. Show that steady state multiplicity is impossible in this case.

Problem 2: Consider a simplified version of the Langmuir-Hinshelwood mechanism analyzed in this section, when both CO and O_2 adsorb irreversibly. This corresponds to $k_{-1} = 0$. In this regime, analyze the dependence of the rate of CO oxidation as function of CO pressure (k_1 in the model). Compare your results quantitatively to the parametric plot of reaction rate obtained with $k_{-1} = 0.05$.

Problem 3: Find all steady states of the model for a specific set of model parameters: $k_1 = 0.5$, $k_{-1} = 0.05$, $k_2 = 1$, and $k_3 = 10$. This can be done numerically in Matlab, using the 'roots' subroutine for solving polynomial equations. Add small random perturbations to each of the four steady states and use the result as an initial condition for the time dependent problem. Plot the results in the (x, y) plane. Perform linear stability analysis of all steady states.

$\boxed{1}$ ①

$$(3+4) \qquad r = \frac{\underbrace{C_T^3 k_3}\left(K_1 K_2 P_{C_2H_4} P_{H_2} - \cancel{\frac{P_{C_2H_6}}{K_3}}\right)}{\left(1 + K_1 P_{C_2H_4} + \sqrt{K_2 P_{H_2}}\right)^3}$$

$\rightarrow 0$ usu.

really, this is $k_1' = k_1 P_{CO}$ It should be possible to lump these.

ethylene $\rightarrow$ ethane

$CO \rightarrow CO_2$

$K_1 = k_1/k_{-1}$ $\quad CO + * \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} CO^*$

$K_2 = k_2/k_{-2}$ $\quad O_2 + 2* \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} 2O^*$ — read as "two adsorbed O"

$K_3 = k_3/k_{-3}$ $\quad CO^* + O^* \underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}} CO_2 + 2*$ — read as "two vacant sites"

Equilibrium of Ad/Desorption

$$K_1 = \frac{[CO^*]}{[*] P_{CO}} \Rightarrow [CO^*] = K_1 P_{CO} [*] \qquad (1)$$

$$K_2 = \frac{[O^*]^2}{[*]^2 P_{O_2}} \Rightarrow [O^*]^2 = K_2 P_{O_2} [*]^2 \qquad (2a)$$

$$[O^*] = \sqrt{K_2 P_{O_2}}\,[*] \qquad (2b)$$

$$C_T = [*] + \underset{(1)}{[CO^*]} + \underset{(2b)}{[O^*]} \qquad (3)$$

total # of catalytic sites is conserved

$$= [*]\left(1 + K_1 P_{CO} + \sqrt{K_2 + P_{O_2}}\right)$$

— C_T is some sort of constant proportional to catalyst area.

product rate

$$r = k_3\left(\underset{(1)}{[CO^*]}\underset{(2b)}{[O^*]} - \frac{P_{CO_2}}{K_3}[*]^2\right) \qquad (4)$$

$$r = k_3 [*]^2\left(K_1 P_{CO}\sqrt{K_2 P_{O_2}} - P_{CO_2}/K_3\right)$$

$\rightarrow 0$ because 1) $CO^* + O^* \rightarrow CO_2$ is irreversible 2) CO_2 is quickly removed

$$\boxed{r = \frac{k_3 C_T^2\left(K_1 P_{CO}\sqrt{K_2 P_{O_2}} - \cancel{P_{CO_2}/K_3}\right)}{\left(1 + K_1 P_{CO} + \sqrt{K_2 + P_{O_2}}\right)^2}}$$

$[CO^*]$ is a fraction $\in (0,1)$

$C_T = 1$

$$= k_3\left(K_1 P_X \sqrt{K_2 P_Y}\right) \Big/ \left(1 + K_1 P_X + \sqrt{K_2 + P_Y}\right)$$

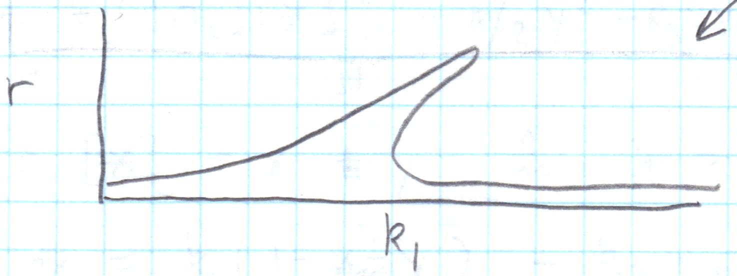
$\boxed{1}$

$$r = k_3\left(k_1 P_x / k_{-1} \sqrt{k_2 P_y / k_{-2}}\right) \Big/ \left(1 + k_1 P_x / k_{-1} + \sqrt{k_2/k_{-2} + P_Y}\right)$$

$$\boxed{r = k_3\left(k_1' / k_{-1} \sqrt{K_2 P_Y}\right) \Big/ \left(1 + k_1'/k_{-1} + \sqrt{K_2 + P_Y}\right)}$$

$$= c_1 \cdot k_1' \Big/ \left(1 + c_2 k_1' + c_3\right) \qquad \boxed{k_1' = k_1 P_{CO}}$$

Because $r = r(k_1')$ is a one-to-one function of k_1' for a given set of pressures, it is not possible to draw a plot like this, & therefore SS-multiplicity is not possible.

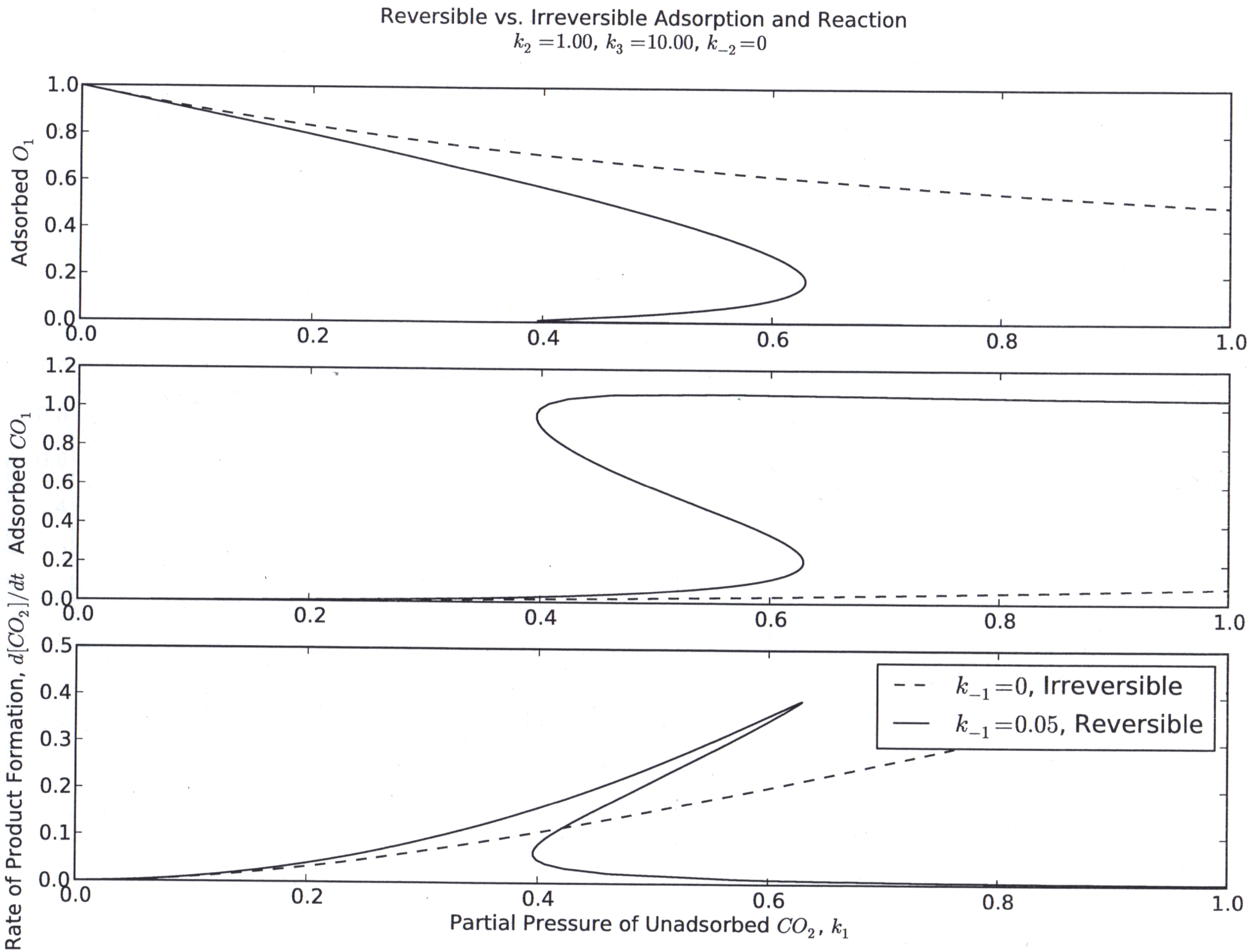


$\boxed{2}$ for comparison, with $k_{-1} = 0.05 \neq 0$

$$\left.\begin{aligned} k_1 &= (k_{-1} + k_3 y)\left(\sqrt{1 + 4k_2(1-y)/k_3 y} - 1\right)/2 \\ x &= k_1(1-y)/(k_1 + k_{-1} + k_3 y) \end{aligned}\right\}$$

Use for parametric plotting

(equations from posted class notes)



Reversible vs. Irreversible Adsorption and Reaction
$k_2 = 1.00$, $k_3 = 10.00$, $k_{-2} = 0$
Adsorbed O_1
Adsorbed CO_1
Rate of Product Formation, $d[CO_2]/dt$
Partial Pressure of Unadsorbed CO_2, k_1
$k_{-1} = 0$, Irreversible
$k_{-1} = 0.05$, Reversible

$\boxed{3}$ 5

$$k_1(1-y) = (k_{-1} + k_3 y + k_1)x$$

$$0 = k_1(1-x-y) - k_{-1}x - k_3 xy$$

$$x = k_1(1-y) \Big/ (k_1 + k_{-1} + k_3 y)$$

$$(1-x-y) = x\cdot(k_{-1}+k_3 y) \Big/ k_1$$

$$0 = k_2(1-x-y)^2 - k_3 xy$$

$$= \frac{k_2}{k_1^2} x^2 (k_{-1}+k_3 y)^2 - k_3 y\, \frac{k_1(1-y)}{(k_1+k_{-1}+k_3 y)}$$

$$= \frac{k_2\, \cancel{k_1^2}(1-y)^2 (k_{-1}+k_3 y)^2}{\cancel{k_1^2}(k_1+k_{-1}+k_3 y)^2} - \frac{k_3 y k_1 (1-y)}{(k_1+k_{-1}+k_3 y)}$$

$$= k_2(1-y)^2(k_{-1}+k_3 y)^2$$
$$- k_3 y k_1(1-y)(k_1+k_{-1}+k_3 y)$$

$$= y^0(k_2 k_{-1}^2) + y^1(-k_3 k_1^2 - k_1 k_3 k_{-1} + 2k_2 k_3 k_{-1} - 2k_2 k_{-1}^2)$$
$$+ y^2(k_1^2 k_3 - k_1 k_3^2 + k_1 k_3 k_{-1} + k_2 k_3^2 - 4k_2 k_3 k_{-1} + k_2 k_{-1}^2)$$
$$+ y^3(k_1 k_3^2 - 2k_2 k_3^2 + 2k_2 k_3 k_{-1}) + y^4(k_2 k_3^2)$$

13 Linear stability analysis

$$f_1(x,y) = \frac{dx}{dt} = k_1(1-x-y) - k_{-1}x - k_3xy$$

$$f_2(x,y) = \frac{dy}{dt} = k_2(1-x-y)^2 - k_3xy$$

$$= k_2(1-x-y-x+x^2+xy-y+xy+y^2) - k_3xy$$

$$= k_2(1-2x-2y+x^2+2xy+y^2) - k_3xy$$

Jacobian

$$\underline{\underline{A}} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix} = \begin{bmatrix} -k_1 - k_{-1} - k_3y & -k_1 - k_3x \\ -2k_2 + 2k_2x + 2k_2y - k_3y & -2k_2 + 2k_2x + 2k_2y - k_3x \end{bmatrix}$$

Steady States: i (0, 1), ii (0.054, 0.451), iii (0.522, 0.037), iv (0.884, 0.001)

$$0 = \left|\underline{A} - \lambda \underline{\underline{I}}\right| = \left(\frac{\partial f^1}{\partial x} - \lambda\right)\cdot\left(\frac{\partial f^2}{\partial y} - \lambda\right) - \left(\frac{\partial f_2}{\partial x}\right)\left(\frac{\partial f_1}{\partial y}\right)\Bigg|_{SS}$$

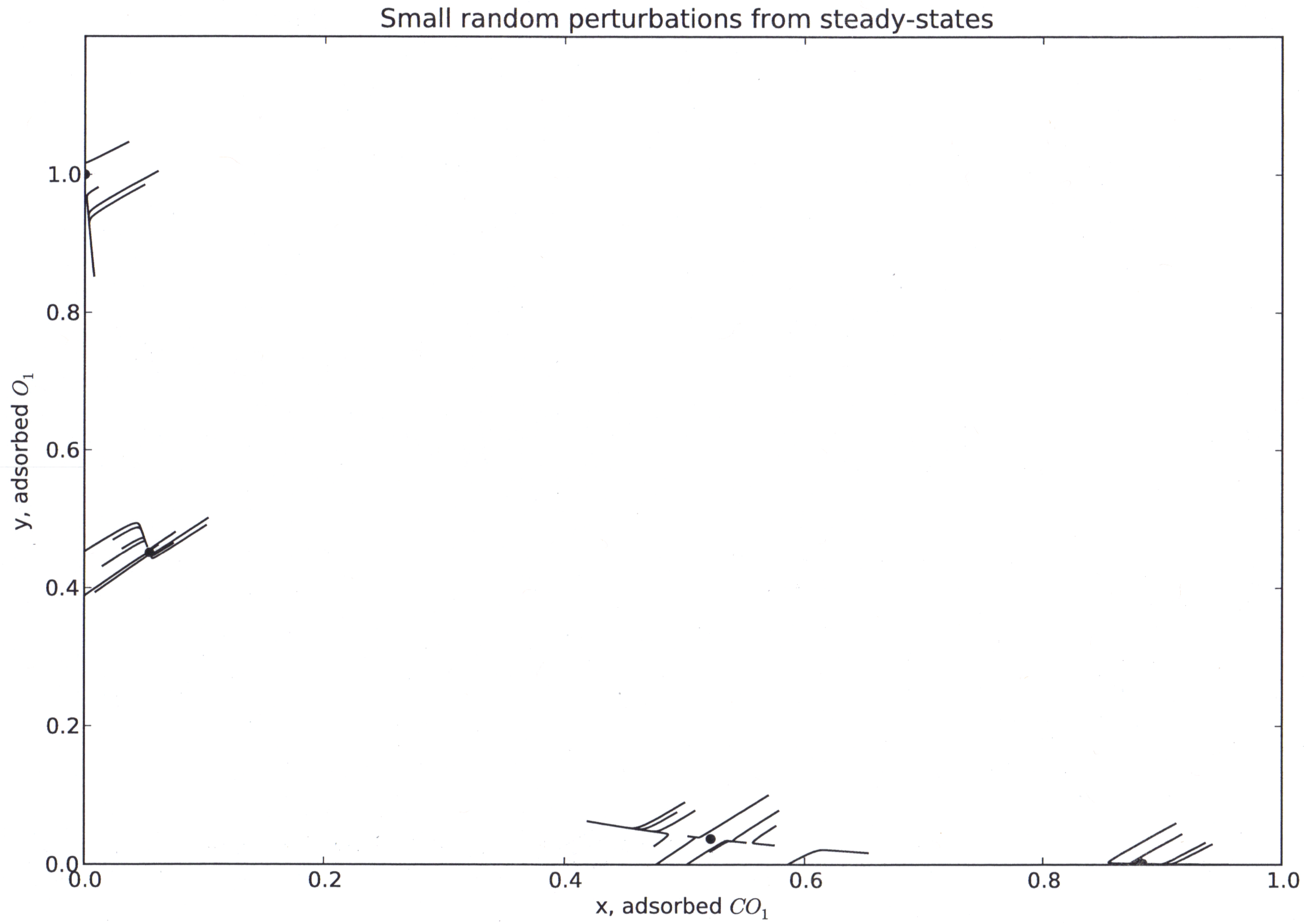
$$\underline{\underline{A}} = \begin{bmatrix} -10.55 & -0.5 \\ -10 & 0 \end{bmatrix} \text{ OR } \begin{bmatrix} -5.06 & -1.04 \\ -5.50 & -1.53 \end{bmatrix} \text{ etc.}$$

if $\underline{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ $\quad 0 = \left|\underline{A} - \lambda\underline{\underline{I}}\right| = (a-\lambda)(d-\lambda) - cb = \lambda^2 + \lambda(-d-a) + (ad-cb)$

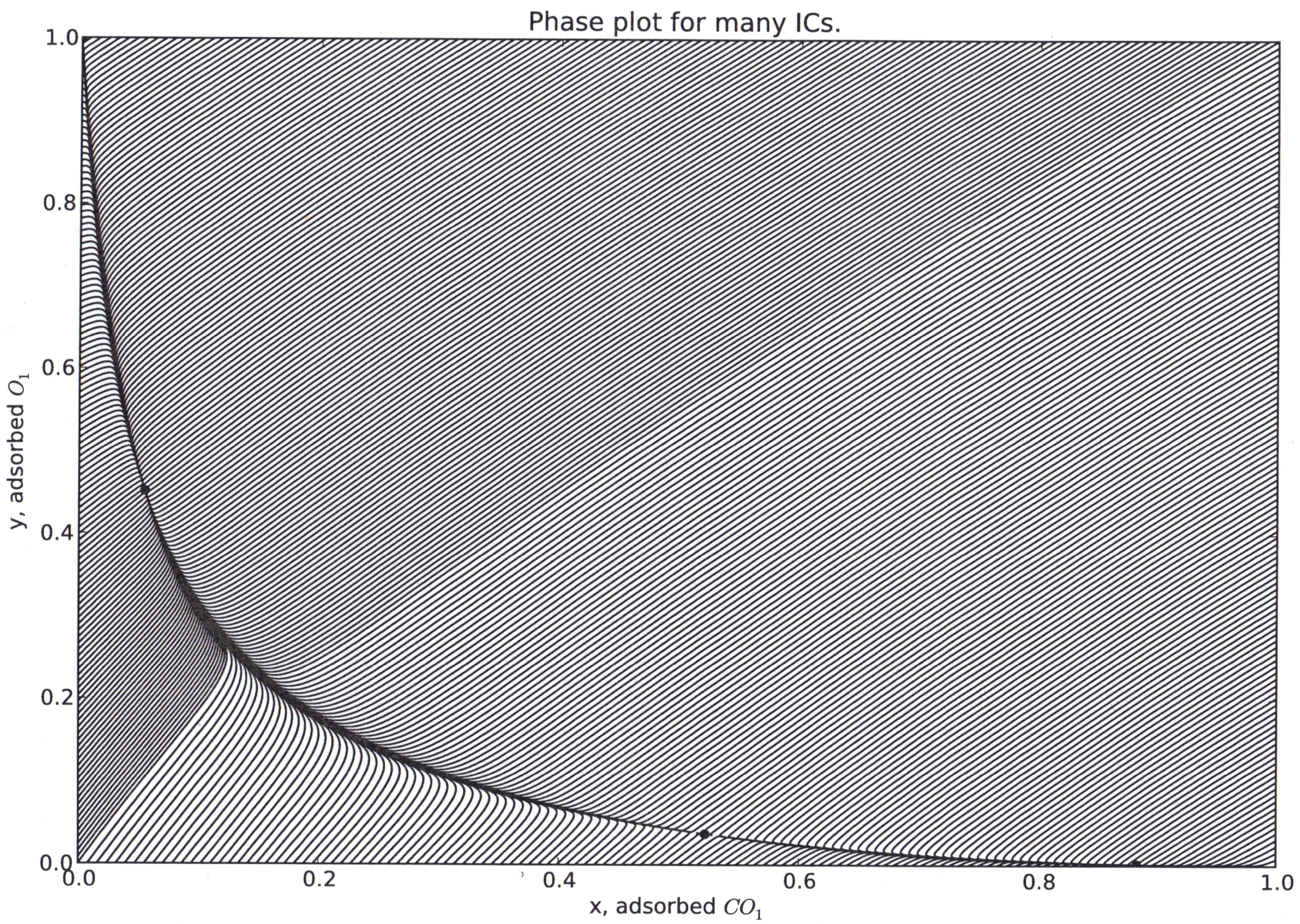
eigenvalues:

$\lambda =$

i	ii	iii	iv
(−11.0, 0.45) *	(−6.27, −0.32)	(−7.32, 0.21) *	(−9.32, −0.305)
UNSTABLE SADDLE	STABLE NODE	UNSTABLE SADDLE	STABLE NODE



Small random perturbations from steady-states
y, adsorbed O_1
x, adsorbed CO_1
0.0
0.2
0.4
0.6
0.8
1.0



Phase plot for many ICs.
y, adsorbed O_1
x, adsorbed CO_1
0.0
0.2
0.4
0.6
0.8
1.0

```python
###############################################################################
################ Code online at github.com/tsbertalan/504hw4 #################
###############################################################################
import numpy as np
import matplotlib.pyplot as plt
from math import sqrt

def do_IC(k1, kn1, k2, kn2, k3, x0, y0, dt=0.01, numsteps=256):
    def xprime(x, y):
        return k1 * (1 - x - y) - kn1 * x - k3 * x * y
    def yprime(x, y):
        return k2 * (1 - x - y)**2 - kn2 * y ** 2 - k3 * x * y
    def pprime(x, y):
        return k3 * x * y

    xold = x0
    yold = y0
    tlist = []
    xlist = []
    ylist = []
    plist = []
    pold = 0
    told = 0
    tlist.append(told)
    xlist.append(xold)
    ylist.append(yold)
    plist.append(pold)
    for j in range(numsteps):
        x = xold + dt * xprime(xold, yold)
        y = yold + dt * yprime(xold, yold)
        p = pold + dt * pprime(xold, yold)
        t = told + dt
        xlist.append(x)
        xold = x
        ylist.append(y)
        yold = y
        plist.append(p)
        pold = p
        tlist.append(t)
        told = t
    return (tlist, xlist, ylist, plist)

def make_ICs(resolution=32):
    ICs = []
    for x0 in np.arange(0, 1. + 1./resolution, 1./resolution):
        ICs.append((x0, 0))
        ICs.append((x0, 1))
        ICs.append((0, x0))
        ICs.append((1, x0))
    return ICs

# Problem 2 ###################################################################
k2 = 1
k3 = 10
fk1 = lambda y: k2*k3*(1-y)/k3/y
fx = lambda y: fk1(y)*(1-y)/(fk1(y)+k3*y)
fr = lambda x,y: k3*x*y
l1 = list(np.arange(0.001,       1,   .001))
l2 = list(np.arange(0.0001,    .001,  .0001))
l3 = list(np.arange(0.00001,  .0001, .00001))
l4 = list(np.arange(0.000001,.00001,.000001))
ylist = l1
ylist.extend(l2)  # We need extra resolution at low y
ylist.extend(l3)  # to properly capture the behavior.
ylist.extend(l4)  # Only a couple dozen extra points.
ylist.sort()
```

```python
xlist = []
k1list= []
rlist = []

kn1 = 0.05
fk1n = lambda y: (kn1+k3*y)*(sqrt(1+4*k2*(1-y)/k3/y)-1)/2
fxn = lambda y: fk1n(y)*(1-y)/(fk1n(y)-kn1+k3*y)
frn = fr
ylistn = []
xlistn = []
k1listn= []
rlistn = []
for y in ylist:
    x = fx(y)
    xlist.append(x)
    k1list.append(fk1(y))
    rlist.append(fr(x,y))

    ylistn.append(y)
    xn = fxn(y)
    xlistn.append(xn)
    k1listn.append(fk1n(y))
    rlistn.append(frn(xn,y))

# Make a plot comparing reversible and reversible
fig2b = plt.figure(22, figsize=(12, 8))

ax2O = fig2b.add_subplot(3,1,1)
ax2O.set_xlim((0,1))
ax2O.plot(k1list, ylist, 'k--')
ax2O.plot(k1listn, ylistn, 'k-')
ax2O.set_ylabel(r'Adsorbed $O_1$')

ax2CO = fig2b.add_subplot(3,1,2)
ax2CO.set_xlim((0,1))
ax2CO.plot(k1list,xlist, 'k--')
ax2CO.plot(k1listn,xlistn, 'k-')
ax2CO.set_ylabel(r'Adsorbed $CO_1$')

ax2r = fig2b.add_subplot(3,1,3)
ax2r.set_xlim((0,1))
ax2r.set_ylim((0,.5))
ax2r.plot(k1list, rlist, 'k--')
ax2r.plot(k1listn, rlistn, 'k-')
ax2r.set_ylabel(r'Rate of Product Formation, ${d[CO_2]}/{dt}$')
ax2r.set_xlabel(r'Partial Pressure of Unadsorbed $CO_2$, $k_1$')

ax2r.legend([r'$k_{-1}=0$, Irreversible', r'$k_{-1}=%.2f$, Reversible'%kn1])
plt.suptitle('Reversible vs. Irreversible Adsorption and Reaction\n'\
             r'$k_2=%.2f$, $k_3=%.2f$, $k_{-2}=0$' % (k2, k3))
plt.savefig('hw4_2b.pdf')
#plt.show()

# Make a plot showing irreversible only, since it has
# qualitatively interesting large-k1 behavior that won't
# fit on the previous plot
fig2c = plt.figure(23, figsize=(12, 8))

ax2O = fig2c.add_subplot(3,1,1)
ax2O.set_xlim((0,100))
ax2O.plot(k1list, ylist, 'k--')
ax2O.set_ylabel(r'Adsorbed $O_1$')

ax2CO = fig2c.add_subplot(3,1,2)
ax2CO.set_xlim((0,100))
ax2CO.plot(k1list,xlist, 'k--')
```

```
ax2CO.set_ylabel(r'Adsorbed $CO_1$')

ax2r = fig2c.add_subplot(3,1,3)
ax2r.set_xlim((0,100))
ax2r.plot(k1list, rlist, 'k--')
ax2r.set_ylabel(r'Rate of Product Formation, ${d[CO_2]}/{dt}$')
ax2r.set_xlabel(r'Partial Pressure of Unadsorbed $CO_2$, $k_1$')

ax2r.legend([r'$k_{-1}=0$, Irreversible'])
plt.suptitle('Irreversible Adsorption and Reaction\n'\
             r'$k_2=%.2f$, $k_3=%.2f$, $k_{-2}=0$' % (k2, k3))
plt.savefig('hw4_2c.pdf')
#plt.show()


# Problem 3 ###################################################################
k1=.5
kn1=.05
k2=1
kn2=0
k3=10
# roots of the cubic polynomial in y that results from setting
#    dx/dt = 0 = dy/dt
# coefficients of said polynomial:

coeffs = [
     k2*k3**2,
     k1*k3**2 - 2*k2*k3**2 + 2*k2*k3*kn1,
     k1**2*k3 - k1*k3**2 + k1*k3*kn1 + k2*k3**2 - 4*k2*k3*kn1 + k2*kn1**2,
     -k3*k1**2 - k1*k3*kn1 +2*k2*k3*kn1 - 2*k2*kn1**2,
     k2*kn1**2
     ]

yroots = np.roots(coeffs)
fx = lambda y: k1*(1-y)/(k1+kn1+k3*y)
xroots = map(fx, yroots)

fig3 = plt.figure(3, figsize=(12, 8))
ax3 = fig3.add_subplot(1,1,1)
ax3.set_xlim((0, 1))
ax3.set_ylim((0, 1.2))
ICs_ss = zip(xroots, yroots)
print '#### Problem 3, steady-states ####'
print "Steady-states (x, y) are:"
for (x, y) in ICs_ss:
    print '(%.3f, %.3f)' % (x, y)
repeats = 12
for (x0,y0) in ICs_ss * repeats:
    x0 = x0 + np.random.rand()*1/8. - 1/16.
    y0 = y0 + np.random.rand()*1/8. - 1/16.
    (tlist, xlist, ylist, plist) = do_IC(k1, kn1, k2, kn2, k3, x0, y0)
    ax3.plot(xlist, ylist, 'k')
for (x0,y0) in ICs_ss:
    ax3.scatter([x0], [y0], color='k')
ax3.set_title('Small random perturbations from steady-states')
ax3.set_xlabel(r'x, adsorbed $CO_1$')
ax3.set_ylabel(r'y, adsorbed $O_1$')
plt.savefig('hw4_3.pdf')

fig3b = plt.figure(32, figsize=(12, 8))
ax3b = fig3b.add_subplot(1,1,1)
ICs = make_ICs(resolution=128)

for (x0,y0) in ICs:
    (tlist, xlist, ylist, plist) = do_IC(k1, kn1, k2, kn2, k3, x0, y0)
    ax3b.plot(xlist, ylist, 'k')
```

```
    ax3b.set_xlim((0,1))
    ax3b.set_ylim((0,1))
for (x0,y0) in ICs_ss:
    ax3b.scatter([x0], [y0], color='k')
ax3b.set_xlim((0, 1))
ax3b.set_ylim((0, 1))
ax3b.set_title('Phase plot for many ICs.')
ax3b.set_xlabel(r'x, adsorbed $CO_1$')
ax3b.set_ylabel(r'y, adsorbed $O_1$')
plt.savefig('hw4_3b.pdf')

# Problem 3, Linear stability analysis #########################################
f1x = lambda x,y: -k1 -kn1 - k3*y
f2x = lambda x,y: -2*k2+2*k2*x+2*k2*y-k3*y
f1y = lambda x,y: -k1-k3*x
f2y = lambda x,y: -2*k2+2*k2*x+2*k2*y-k3*x
print ''
print '#### Problem 3, Linear stability analysis ####'
for (x,y) in ICs_ss:
    jacobian = np.array([[f1x(x,y), f1y(x,y)],
                         [f2x(x,y), f2y(x,y)]])
    print ''
    print 'At (', x, ',', y, '), the Jacobian is:'
    print jacobian
    [a,b,c,d] = jacobian.flatten().tolist()
    coeffs = [1, -d-a, a*d-c*b]
    eigv = np.roots(coeffs)
    print 'Eigenvalues are:', eigv
```