Digital simulation of CV: 1D case

Excerpt from Prof. Kenji Kano's book on basic electrochemisrty

Modified on February 6, 2009 10 : 19 pm

1 Diffusion equation

The Fick's first and second law for diffusion and the differential equation for diffusion are

$$\vec{\mathbf{J}}(\mathbf{r},t) = -D\nabla c(\mathbf{r},t) \tag{1}$$

$$\frac{\partial c}{\partial t} + \mathrm{div}\cdot\vec{\mathbf{J}}(\mathbf{r},t) = 0 \tag{2}$$

$$\frac{\partial c}{\partial t} = D\nabla^2 C \tag{3}$$

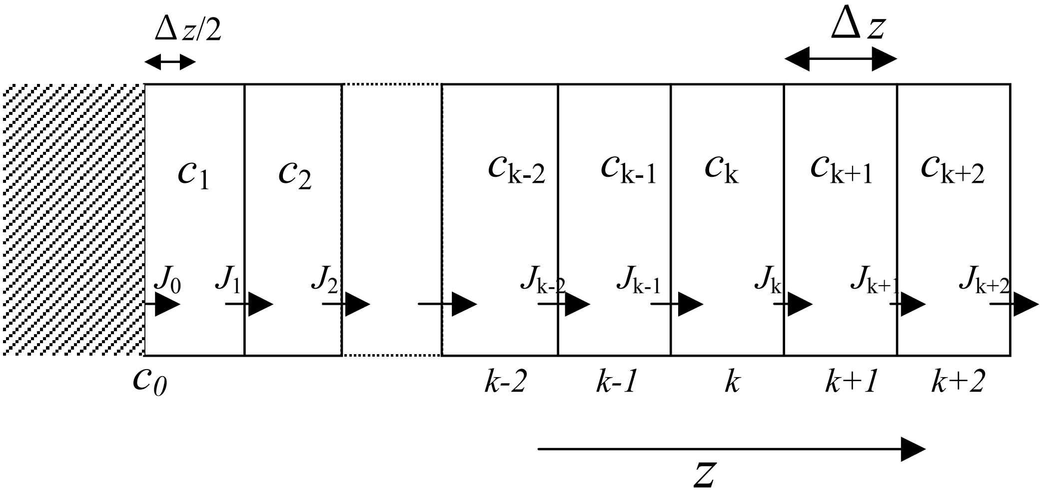

Figure 1: Discrete model of diffusion

If we consider one-dimensional diffusion, and the above equations are approximated by discretized model for numerical simulation

$$J_k(t) = -D\frac{c_{k+1}(t)-c_k(t)}{\Delta z} \tag{4}$$

$$\begin{aligned}
\frac{c_k(t+\Delta t)-c_k(t)}{\Delta t}\Delta x\Delta y\Delta z &= [J_{k-1}(t)-J_k(t)]\Delta x\Delta y \\
\frac{c_k(t+\Delta t)-c_k(t)}{\Delta t} &= \frac{J_{k-1}(t)-J_k(t)}{\Delta z} \\
&= \frac{D}{(\Delta z)^2}[c_{k+1}(t)-2c_k(t)+c_{k-1}(t)]
\end{aligned} \tag{5}$$

$$c_k(t+\Delta t) = c_k(t) + D_M[c_{k+1}(t)-2c_k(t)+c_{k-1}(t)] \tag{6}$$

$$D_M = \frac{D\Delta t}{(\Delta z)^2} \tag{7}$$

$$c_k(t+\Delta t) = \begin{pmatrix} D_M & 1-2D_M & D_M \end{pmatrix}\begin{pmatrix} c_{k+1}(t) \\ c_k(t) \\ c_{k-1}(t) \end{pmatrix} \tag{8}$$

D_M is the dimensionless and used 0.45 (< 0.5) for a stable calculation. For $k = 1$ and $k = N$(final mesh point) we have

$$\frac{c_1(t+\Delta t)-c_1(t)}{\Delta t} = \frac{J_0(t)-J_1(t)}{\Delta z}, \quad J_0(t) = -D\frac{c_1(t)-c_0(t)}{\Delta z/2}, J_1(t) = -D\frac{c_2(t)-c_1(t)}{\Delta z}$$

$$\begin{aligned}
\frac{c_N(t+\Delta t)-c_N(t)}{\Delta t} &= \frac{J_{N-1}(t)-J_N(t)}{\Delta z}, \quad J_{N-1}(t)=-D\frac{c_N(t)-c_{N-1}(t)}{\Delta z}, J_N(t)=0 \\
c_1(t+\Delta t) &= c_1(t)+D_M[2c_0(t)-3c_1(t)+c_2(t)] \qquad (9) \\
c_N(t+\Delta t) &= c_N(t)+D_M[c_{N-1}(t)-c_N(t)] \qquad (10)
\end{aligned}$$

If we use the matrix

$$\begin{pmatrix} c_0(t+\Delta t) \\ c_1(t+\Delta t) \\ c_2(t+\Delta t) \\ ... \\ ... \\ c_{N-1}(t+\Delta t) \\ c_N(t+\Delta t) \end{pmatrix} = \begin{pmatrix} ? & ? & 0 & ... & & & & \\ 2D_M & 1-3D_M & D_M & 0 & ... & & & \\ 0 & D_M & 1-2D_M & D_M & 0 & .. & & \\ ... & & & & & & & \\ ... & & & & & & & \\ 0 & 0 & 0 & ... & 0 & D_M & 1-2D_M & D_M \\ 0 & 0 & 0 & ... & 0 & 0 & D_M & 1-D_M \end{pmatrix} \begin{pmatrix} c_0(t) \\ c_1(t) \\ c_2(t) \\ ... \\ ... \\ c_{N-1}(t) \\ c_N(t) \end{pmatrix}$$

We will use

$$\begin{aligned}
E(t) &= \begin{cases} E_0 \mp vt, & 0<t<t_h \\ E_0 \mp 2vt_h \pm vt, & t_h<t<2t_h \end{cases} \qquad (11) \\
\Delta E &= E_{\max}/i_{\max} = 2vt_h/i_{\max} \qquad (12) \\
\Delta t &= t_{\max}/i_{\max} = 2t_h/i_{\max} \qquad (13) \\
t(i) &= i\Delta t \qquad (14) \\
\Delta z &= \sqrt{\frac{D\Delta t}{D_M}} \qquad (15) \\
z(k) &= k\Delta z \qquad (16) \\
z_{\max} &= 6\sqrt{Dt_h}\ (>\sqrt{\pi D t_h}) \qquad (17)
\end{aligned}$$

Eaxample: $v = 50$ mV s^{-1}, $E_{\max} = 2\times 500$ mV, $2t_h$= 20 s, $i_{\max} = 10000, \Delta t = 2\times 10^{-3}$ s, $\Delta E = 0.1$ mV, $D_M = 0.45$, $D = 10^{-9}$ m^2 s^{-1}, $\Delta z = 2\times 10^{-6}$ m, $z_{\max} = 6\times 10^{-4}$ m, $k_{\max} = 300$

I don't like dimensionless simulation, but, it is convenient to use dimensionless concentration unit $f = c/c^*$.

$$f_k(t+\Delta t) = f_k(t) + D_M[f_{k+1}(t) - 2f_k(t) + f_{k-1}(t)] \qquad (18)$$

Here the position index k, the time index i, and the concentration f and the diffusion constant D_M are all dimensionless parameters. For the mesh just above the electrode surface ($k = 1$)

$$f_1(t+\Delta t) = f_1(t) + D_M[f_2(t) - f_1(t)] - J_{k=0}\Delta t/c^*\Delta z \qquad (19)$$

2 Initial condition

The initial potential E_0 is chosen for the condition that there is no current $i = 0$. This means that the initial potential E_0 is much positive(negative) than the formal redox potential $E^{\ominus'}$ of for the reduction(oxidation).

$$\begin{aligned}
&\text{Reduction} : E_0 > E^{\ominus'} \qquad (20) \\
c_{\text{Ox}}(z, t=0) &= c^*, \quad c_{\text{Rd}}(z, t=0) = 0, \qquad (21) \\
&\text{Oxidation} : E_0 < E^{\ominus'} \qquad (22) \\
c_{\text{Ox}}(z, t=0) &= 0, \quad c_{\text{Rd}}(z, t=0) = c^* \qquad (23)
\end{aligned}$$

3 Bounday condition:Redox at the electrode

In the bulk at $z = z_{\max}$

$$\begin{aligned}
&\text{Reduction} : E_0 > E^{\ominus'} \qquad (24) \\
c_{\text{Ox}}(z_{\max}, t) &= c^*, \quad c_{\text{Rd}}(z_{\max}, t) = 0, \qquad (25) \\
&\text{Oxidation} : E_0 < E^{\ominus'} \qquad (26) \\
c_{\text{Ox}}(z_{\max}, t) &= 0, \quad c_{\text{Rd}}(z_{\max}, t) = c^* \qquad (27)
\end{aligned}$$

Now we consider the redox at the electorde

$$\text{Ox}^z + ne^- \rightleftharpoons \text{Rd}^{z-n} \qquad (28)$$

The diffusion equations for O and R are given

$$\frac{\partial c_{\mathrm{Ox}}}{\partial t} = D_{\mathrm{Ox}} \frac{\partial^2 c_{\mathrm{Ox}}}{\partial z^2} \tag{29}$$

$$\frac{\partial c_{\mathrm{Rd}}}{\partial t} = D_{\mathrm{Rd}} \frac{\partial^2 c_{\mathrm{Rd}}}{\partial z^2} \tag{30}$$

The Faraday current i_f and i_b is given by

$$\frac{i_f(t)}{nFA} = -k_f c_{\mathrm{Ox}}(z=0,t) \tag{31}$$

$$\frac{i_b(t)}{nFA} = +k_b c_{\mathrm{Rd}}(z=0,t) \tag{32}$$

The reductive current is defined by negative (e.g. for reductive desorption for Au-S-$(CH_2)_nCH_3$ + e^- → Au + (S-$(CH_2)_nCH_3$), then the minus sign for $i_f(t)$. The dimension of the lhs of the above equations are the same as flux [A / (C mol^{-1} m^2)] = [C s^{-1} / (C mol^{-1} m^2)] = [mol m^{-2} s^1], then the dimension of k_f and k_b is [m s^{-1}].

The net current density i becomes

$$\frac{i}{nFA} = \frac{i_f + i_b}{nFA} = -k_f c_{\mathrm{Ox}}(z=0,t) + k_b c_{\mathrm{Rd}}(z=0,t) \tag{33}$$

$$\tag{34}$$

If we use the Butler-(Erdey-Grùz)-Volmer relation, we can get

$$k_f(t) = k_0 \exp[-\frac{\alpha n F}{RT}(E(t) - E^{\ominus\prime})] \tag{35}$$

$$k_b(t) = k_0 \exp[\frac{(1-\alpha) n F}{RT}(E(t) - E^{\ominus\prime})] \tag{36}$$

In the case that k_0 is great enough, the system becomes "reversible".

The change of the concentrations O and R at the electrode may be given by

$$\frac{\partial c_{\mathrm{Ox}}(z=0,t)}{\partial t} dV = [-J_{\mathrm{Ox}}(z=0,t) - k_f(t) c_{\mathrm{Ox}}(z=0,t) + k_b(t) c_{\mathrm{Rd}}(z=0,t)] dS \tag{37}$$

$$\frac{\partial c_{\mathrm{Rd}}(z=0,t)}{\partial t} dV = [-J_{\mathrm{Rd}}(z=0,t) - k_b(t) c_{\mathrm{Rd}}(z=0,t) + k_f(t) c_{\mathrm{Ox}}(z=0,t)] dS \tag{38}$$

$$dV = \Delta x \Delta y \Delta z_{\mathrm{surface}}, \quad dS = \Delta x \Delta y \tag{39}$$

Hereafter we use the definition for $c_{\mathrm{Ox}}(z,t)$ and $c_{\mathrm{Rd}}(z,t)$

$$c_{\mathrm{Ox}}(z,t) \equiv c_{\mathrm{Ox}}(i,t), \quad c_{\mathrm{Rd}}(z,t) \equiv c_{\mathrm{Rd}}(i,t); \quad z(i) = z \tag{40}$$

The discredized form of the above equations are

$$J_{\mathrm{Ox}}(z=0,t) = -\frac{2D_{\mathrm{Ox}}}{\Delta z}[c_{\mathrm{Ox}}(1,t) - c_{\mathrm{Ox}}(0,t)] \tag{41}$$

$$J_{\mathrm{Rd}}(z=0,t) = -\frac{2D_{\mathrm{Rd}}}{\Delta z}[c_{\mathrm{Rd}}(1,t) - c_{\mathrm{Rd}}(0,t)] \tag{42}$$

$$\frac{c_{\mathrm{Ox}}(0,t+\Delta t) - c_{\mathrm{Ox}}(0,t)}{\Delta t} \Delta z_{\mathrm{surface}} = \frac{2D_{\mathrm{Ox}}}{\Delta z}[c_{\mathrm{Ox}}(1,t) - c_{\mathrm{Ox}}(0,t)] - k_f(t) c_{\mathrm{Ox}}(0,t) + k_b(t) c_{\mathrm{Rd}}(0,t) \tag{43}$$

$$\frac{c_{\mathrm{Rd}}(0,t+\Delta t) - c_{\mathrm{Rd}}(0,t)}{\Delta t} \Delta z_{\mathrm{surface}} = \frac{2D_{\mathrm{Rd}}}{\Delta z}[c_{\mathrm{Rd}}(1,t) - c_{\mathrm{Rd}}(0,t)] - k_b(t) c_{\mathrm{Rd}}(0,t) + k_f(t) c_{\mathrm{Ox}}(0,t) \tag{44}$$

In the limit $\Delta z_{\mathrm{surface}} \to 0$, the lhs of Eq.(43) and Eq.(44) become zero, then from Eq.(43) + Eq.(44)

$$D_{\mathrm{Ox}}[c_{\mathrm{Ox}}(1,t) - c_{\mathrm{Ox}}(0,t)] = -D_{\mathrm{Rd}}[c_{\mathrm{Rd}}(1,t) - c_{\mathrm{Rd}}(0,t)] \tag{45}$$

This means that the fluxs of Ox and Red are cancelled out. From Eq.(43) and Eq.(44)

$$\left[\frac{2D_{\mathrm{Ox}}}{\Delta z} + k_f(t)\right] c_{\mathrm{Ox}}(0,t) = \frac{2D_{\mathrm{Ox}}}{\Delta z} c_{\mathrm{Ox}}(1,t) + k_b(t) c_{\mathrm{Rd}}(0,t) \tag{46}$$

$$c_{\mathrm{Rd}}(0,t) = \frac{\frac{2D_{\mathrm{Rd}}}{\Delta z} c_{\mathrm{Rd}}(1,t) + k_f(t) c_{\mathrm{Ox}}(0,t)}{\frac{2D_{\mathrm{Rd}}}{\Delta z} + k_b(t)} \tag{47}$$

$$
\begin{aligned}
\left[\frac{2D_{\mathrm{Ox}}}{\Delta z}+k_f(t)\right]c_{\mathrm{Ox}}(0,t) &= \frac{2D_{\mathrm{Ox}}}{\Delta z}c_{\mathrm{Ox}}(1,t)+k_b(t)\frac{\frac{2D_{\mathrm{Rd}}}{\Delta z}c_{\mathrm{Rd}}(1,t)+k_f(t)c_{\mathrm{Ox}}(0,t)}{\frac{2D_{\mathrm{Rd}}}{\Delta z}+k_b(t)} && (48)\\
\left[\frac{2D_{\mathrm{Ox}}}{\Delta z}+k_f(t)-\frac{k_f(t)k_b(t)}{\frac{2D_{\mathrm{Rd}}}{\Delta z}+k_b(t)}\right]c_{\mathrm{Ox}}(0,t) &= \frac{2D_{\mathrm{Ox}}}{\Delta z}c_{\mathrm{Ox}}(1,t)+\frac{k_b(t)\frac{2D_{\mathrm{Rd}}}{\Delta z}}{\frac{2D_{\mathrm{Rd}}}{\Delta z}+k_b(t)}c_{\mathrm{Rd}}(1,t) && (49)\\
c_{\mathrm{Ox}}(0,t) &= B_1c_{\mathrm{Ox}}(1,t)+B_2c_{\mathrm{Rd}}(1,t) && (50)\\
B_1 &= \frac{\frac{2D_{\mathrm{Ox}}}{\Delta z}}{\frac{2D_{\mathrm{Ox}}}{\Delta z}+k_f(t)-\frac{k_f(t)k_b(t)}{\frac{2D_{\mathrm{Rd}}}{\Delta z}+k_b(t)}} && (51)\\
B_2 &= \frac{\frac{k_b(t)\frac{2D_{\mathrm{Rd}}}{\Delta z}}{\frac{2D_{\mathrm{Rd}}}{\Delta z}+k_b(t)}}{\frac{2D_{\mathrm{Ox}}}{\Delta z}+k_f(t)-\frac{k_f(t)k_b(t)}{\frac{2D_{\mathrm{Rd}}}{\Delta z}+k_b(t)}} && (52)\\
\left[\frac{2D_{\mathrm{Rd}}}{\Delta z}+k_b(t)\right]c_{\mathrm{Rd}}(0,t) &= \frac{2D_{\mathrm{Rd}}}{\Delta z}c_{\mathrm{Rd}}(1,t)+k_f(t)c_{\mathrm{Ox}}(0,t) && (53)\\
&= \frac{2D_{\mathrm{Rd}}}{\Delta z}c_{\mathrm{Rd}}(1,t)+k_f(t)[B_1c_{\mathrm{Ox}}(1,t)+B_2c_{\mathrm{Rd}}(1,t)] && (54)\\
c_{\mathrm{Rd}}(0,t) &= \frac{\frac{2D_{\mathrm{Rd}}}{\Delta z}+k_f(t)B_2}{\frac{2D_{\mathrm{Rd}}}{\Delta z}+k_b(t)}c_{\mathrm{Rd}}(1,t)+\frac{k_f(t)B_1}{\frac{2D_{\mathrm{Rd}}}{\Delta z}+k_b(t)}c_{\mathrm{Ox}}(1,t) && (55)
\end{aligned}
$$

Using Eq.(50, 51, 52, 55) $c_{\mathrm{Ox}}(0,t)$ and $c_{\mathrm{Rd}}(0,t)$ can be determined from $c_{\mathrm{Ox}}(1,t), c_{\mathrm{Rd}}(1,t), k_f(t), k_b(t)$.

If we use the matrix and $D_M^{\mathrm{Ox}} = (D_{\mathrm{Ox}}\Delta t)/(\Delta z)^2, D_M^{\mathrm{Rd}} = (D_{\mathrm{Rd}}\Delta t)/(\Delta z)^2$

$$
\begin{pmatrix}
c_{\mathrm{Ox}}(1,t+\Delta t)-2D_M^{\mathrm{Ox}}c_{\mathrm{Ox}}(0,t)\\
c_{\mathrm{Ox}}(2,t+\Delta t)\\
c_{\mathrm{Ox}}(3,t+\Delta t)\\
...\\
...\\
c_{\mathrm{Ox}}(N-1,t+\Delta t)\\
c_{\mathrm{Ox}}(N,t+\Delta t)
\end{pmatrix}
= \qquad (56)
$$

$$
\begin{pmatrix}
1-3D_M^{\mathrm{Ox}} & D_M^{\mathrm{Ox}} & 0 & ... & & & & \\
D_M^{\mathrm{Ox}} & 1-2D_M^{\mathrm{Ox}} & D_M^{\mathrm{Ox}} & 0 & ... & & & \\
0 & D_M^{\mathrm{Ox}} & 1-2D_M^{\mathrm{Ox}} & D_M^{\mathrm{Ox}} & 0 & ... & & \\
... & & & & & & & \\
... & & & & & & & \\
0 & 0 & 0 & ... & 0 & D_M^{\mathrm{Ox}} & 1-2D_M^{\mathrm{Ox}} & D_M^{\mathrm{Ox}}\\
0 & 0 & 0 & ... & 0 & 0 & D_M^{\mathrm{Ox}} & 1-D_M^{\mathrm{Ox}}
\end{pmatrix}
\begin{pmatrix}
c_{\mathrm{Ox}}(1,t)\\
c_{\mathrm{Ox}}(2,t)\\
c_{\mathrm{Ox}}(3,t)\\
...\\
...\\
c_{\mathrm{Ox}}(N-1,t)\\
c_{\mathrm{Ox}}(N,t)
\end{pmatrix}
$$

$$
\begin{pmatrix}
c_{\mathrm{Rd}}(1,t+\Delta t)-2D_M^{\mathrm{Rd}}c_{\mathrm{Rd}}(0,t)\\
c_{\mathrm{Rd}}(2,t+\Delta t)\\
c_{\mathrm{Rd}}(3,t+\Delta t)\\
...\\
...\\
c_{\mathrm{Rd}}(N-1,t+\Delta t)\\
c_{\mathrm{Rd}}(N,t+\Delta t)
\end{pmatrix}
= \qquad (57)
$$

$$
\begin{pmatrix}
1-3D_M^{\mathrm{Rd}} & D_M^{\mathrm{Rd}} & 0 & ... & & & & \\
D_M^{\mathrm{Rd}} & 1-2D_M^{\mathrm{Rd}} & D_M^{\mathrm{Rd}} & 0 & ... & & & \\
0 & D_M^{\mathrm{Rd}} & 1-2D_M^{\mathrm{Rd}} & D_M^{\mathrm{Rd}} & 0 & ... & & \\
... & & & & & & & \\
... & & & & & & & \\
0 & 0 & 0 & ... & 0 & D_M^{\mathrm{Rd}} & 1-2D_M^{\mathrm{Rd}} & D_M^{\mathrm{Rd}}\\
0 & 0 & 0 & ... & 0 & 0 & D_M^{\mathrm{Rd}} & 1-D_M^{\mathrm{Rd}}
\end{pmatrix}
\begin{pmatrix}
c_{\mathrm{Rd}}(1,t)\\
c_{\mathrm{Rd}}(2,t)\\
c_{\mathrm{Rd}}(3,t)\\
...\\
...\\
c_{\mathrm{Rd}}(N-1,t)\\
c_{\mathrm{Rd}}(N,t)
\end{pmatrix}
$$

If the concentration at $z = z_{\max}$ is always the same, this means

$$
\begin{aligned}
c_{\mathrm{Ox}}(N-1,t) &= c_{\mathrm{Ox}}(N,t) && (58)\\
c_{\mathrm{Rd}}(N-1,t) &= c_{\mathrm{Rd}}(N,t) && (59)
\end{aligned}
$$

4 Fortran programs and some examples

You don't need to buy the CV simulation software. You can write the program. In the following we show the **125-lines-only Fortran program** to simulate Butler-Volmer type CVs.

```fortran
c23456789| Source code wrtten by M. Yamamoto EHCC Kyoto-U.
c     CV simulation :unit MKS
c     Butler-Volmer type :reversible q-rev. irrev CVs can simulate
      implicit real*8 (a-h,o-z)
      double precision :: cox(0:10000),crd(0:10000),z(0:10000)
      double precision :: cox_new(0:10000),crd_new(0:10000)
c------ input parameter here
      dox=1.0d-9  ! diffusion const. of Ox  m2 s-1
      drd=1.0d-9  ! diffusion const. of Rd  m2 s-1
      e0=0.25d0   ! start potential V
      v=-0.05d0   ! sweep rate    V s-1;  v < 0:redeuction
      th=10.0d0  !   half time of sweep
      em=e0+v*th  ! E(t)=e0 + v*t      0<= t <= th,
      ef=e0+2.0d0*v*th-2.0d0*v*th     !  E(t)= e0+2*v*th-v*t   th <= t <= 2*th
      eoxrd=0.0d0 ! formal potential of Ox-Red : V
      imax=20000  ! time mesh
      emax=2.0d0*(e0-em) ! full scale of E:V
      tmax=2.0d0*th  ! time to sweep
      delta_t=tmax/real(imax) ! time step in s
      dm=0.45d0   ! step parameter usually taken as 0.45
      dmin=min(dox,drd)       !   smaller diffusion constant
      dmax=max(dox,drd)     !  greater diffusion constant
      delta_z=sqrt(dox*delta_t/dm) ! mesh in z coor. in m
      zmax=6.0d0*sqrt(dmax*th)   ! max in z :   most outer plane in solution
      kmax=int(zmax/delta_z)+1    ! # of mesh in z
      cox0=0.01d0*1000.0d0    ! c_Ox in bulk / mol m-3 be careful 1 M = 1 mol dm-3 = 1000 mol m-3
      crd0=0.0d0                 ! c_Rd in bulk / mol m-3
      ek0=1.0d-5      !k0 / m s-1
      alpha=0.5d0      !alpha parameter in Butlet-Volmer Eq.
      temp=300.0d0  !temp / K
      en=1.0d0      !n: Ox(z) + ne = Rd(z-n)   number of electron
c---------------------- print parameters
      write (6,*) 'dox: diff. const. of Ox in m2 s-1',dox
      write (6,*) 'drd: diff. const. of Rd in m2 s-1',drd
      write (6,*) 'e0: start potential in V',e0
      write (6,*) 'v:sweep rate in V s-1; v < 0:redeuction', v
      write (6,*) 'th: in s; E(t)=e0 + v*t 0<= t <= th',th
      write (6,*) 'em=e0+v*th, ef=e0',em,ef
      write (6,*) 'eoxrd: formal potential of Ox-Red in V',eoxrd
      write (6,*) 'imax: # of time mesh',imax
      write (6,*) 'emax: full scale of E in V',emax
      write (6,*) 'tmax: time to sweep in s',tmax
      write (6,*) 'delta_t: time step in s', delta_t
      write (6,*) 'dm: step parameter usually taken as 0.45', dm
      write (6,*) 'dmin=min(dox,drd),dmax=max(dox,drd)',dmin,dmax
      write (6,*) 'delta_z:mesh in z coor. in m',delta_z
      write (6,*) 'zmax:  max in z / m ',zmax
      write (6,*) 'kmax:  # of mesh in z',kmax
      write (6,*) 'cox0: 1000*c_Ox / M in bulk ',cox0
      write (6,*) 'crd0: 1000*c_Rd / M  in bulk ',crd0
      write (6,*) 'ek0: rate const. k0 / m s-1',ek0
      write (6,*) 'alpha:alpha parameter in Butlet-Volmer Eq.',alpha
      write (6,*) 'temp:temp / K',temp
      write (6,*) 'en: n: Ox(z) + ne = Rd(z-n)', en
c     ------- physical const.
      f=96485.3399d0 ! Faraday const.
```

```
      r=8.314472d0   ! molar gas const.
c     --------- diff coeff. without dimension
      dmox=dox*delta_t/delta_z/delta_z
      dmrd=drd*delta_t/delta_z/delta_z
      write (6,*) 'dmox,dmrd',dmox,dmrd
c     ---- z cooddinate
      do k=0, kmax
       z(k)=real(k)*delta_z
      enddo
c     ---- initial condition ---
      do k=1, kmax
       cox(k)=cox0
       crd(k)=crd0
      enddo
c     ------- surface concentration
      et=e0
      ekft=ek0*exp(-alpha*en*f/r/temp*(et-eoxrd))
      ekbt=ek0*exp((1.0d0-alpha)*en*f/r/temp*(et-eoxrd))
      b1=2.0d0*dox/delta_z/(2.0d0*dox/delta_z+ekft-
     &   ekft*ekbt/(2.0d0*drd/delta_z+ekbt))
      b2=ekbt*2.0d0*drd/delta_z/(2.0d0*drd/delta_z+ekbt)/
     &   (2.0d0*dox/delta_z+ekft-ekft*ekbt/(2.0d0*drd/delta_z+ekbt))
      cox(0)=b1*cox(1)+b2*crd(1)
      crd(0)=(2.0d0*drd/delta_z+ekft*b2)/(2.0d0*drd/delta_z+ekbt)
     &       *crd(1)+ekft*b1/(2.0d0*drd/delta_z+ekbt)*cox(1)
      write (6,*) 'b1,b2,cox(0),crd(0)',b1,b2,cox(0),crd(0)
      write (6,*) 'time, potential / V, current density / Am-2,
     &           cox(surface), crd(surface),cox(1),crd(1),ekft,ekbt'
c     -------- time routine i=1, imax
      do i=1, imax
       time=delta_t*real(i)
       if (time <= th ) then
        et=e0+v*time
       else
        et=e0+2.0d0*v*th-v*time
       endif
       ekft=ek0*exp(-alpha*en*f/r/temp*(et-eoxrd))
       ekbt=ek0*exp((1.0d0-alpha)*en*f/r/temp*(et-eoxrd))
       cox_new(1)=2.0d0*dmox*cox(0)+(1.0d0-3.0d0*dmox)*cox(1)
     &            +dmox*cox(2)
       crd_new(1)=2.0d0*dmrd*crd(0)+(1.0d0-3.0d0*dmrd)*crd(1)
     &            +dmrd*crd(2)
       do k=2, kmax-1
        cox_new(k)=dmox*cox(k-1)+(1.0d0-2.0d0*dmox)*cox(k)
     &             +dmox*cox(k+1)
        crd_new(k)=dmrd*crd(k-1)+(1.0d0-2.0d0*dmrd)*crd(k)
     &             +dmrd*crd(k+1)
       enddo

       b1=2.0d0*dox/delta_z/(2.0d0*dox/delta_z+ekft-
     &   ekft*ekbt/(2.0d0*drd/delta_z+ekbt))
       b2=ekbt*2.0d0*drd/delta_z/(2.0d0*drd/delta_z+ekbt)/
     &   (2.0d0*dox/delta_z+ekft-ekft*ekbt/(2.0d0*drd/delta_z+ekbt))

       cox_new(0)=b1*cox_new(1)+b2*crd_new(1)
       crd_new(0)=(2.0d0*drd/delta_z+ekft*b2)/(2.0d0*drd/delta_z+ekbt)
     &       *crd_new(1)+ekft*b1/(2.0d0*drd/delta_z+ekbt)*cox_new(1)

       cox_new(kmax)=cox0
       crd_new(kmax)=crd0
       current_dens=en*f*(-ekft*cox_new(0)+ekbt*crd_new(0))
```

```
       write (6,'(9e15.7)') time,et,current_dens,cox_new(0),crd_new(0)
     &                      ,cox_new(1),crd_new(1),ekft,ekbt
       do k=0, kmax
        cox(k)=cox_new(k)
        crd(k)=crd_new(k)
       enddo

      enddo ! i loop
      stop
      end
```

Simulation is done by the following condition. dox: diff. const. of Ox in m2 s-1 1.0E-009, drd: diff. const. of Rd in m2 s-1 1.0E-009, e0: start potential in V 0.25, v:sweep rate in V s-1; $v < 0$:redeuction -5.0E-002, th: in s; $E(t) = e0 + v * t0 <= t <= th$ 10.0, em=e0+v*th, ef=e0 -0.25, 0.250, eoxrd: formal potential of Ox-Red in V 0.0, imax: number of time mesh 20000, emax: full scale of E in V 1.0, tmax: time to sweep in s 20.0, delta t: time step in s 1.0E-003, dm: step parameter usually taken as 0.45, delta z:mesh in z coor. in m 1.4907E-006, zmax: max in z / m 6.0E-004, kmax: number of mesh in z 403, cox0: 1000*c Ox in bulk / M 10.0, crd0: 1000*c Rd in bulk / M 0.0, ek0: rate const. k0 / m s-1 form 100 to 1.0E-007, alpha:alpha parameter in Butlet-Volmer Eq. 0.5, temp:temp / K 300.0, en: n: Ox(z) + ne = Rd(z-n) 1.0.

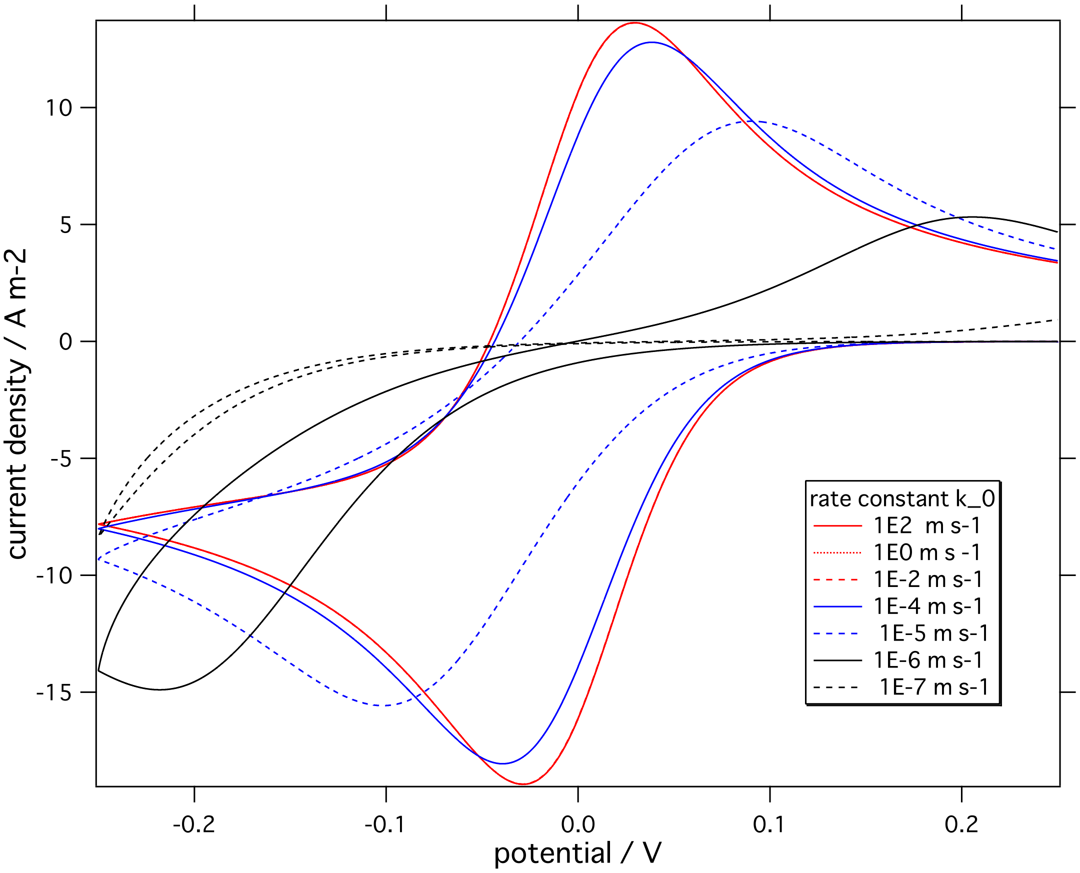

Figure 2: Simulation of CVs for Butler-Volmer rate processes

We checked the program by Nicholson's plot Ψ and peak separation [Anal. Chem. 37, 1351-1355(1965)], where Ψ is given by

$$\Psi = \frac{(D_{\mathrm{Ox}}/D_{\mathrm{Rd}})^{\alpha/2} k_0^{\mathrm{app}}}{\sqrt{\pi D_{\mathrm{Ox}} n F v/(RT)}} \tag{60}$$

As shown in the figures below, the program may work well.

Please remember that when $k_0 \geq 10^{-2}$ m s^{-1} = 1 cm s^{-1}, the CV looks like the reversible one under the experimental conditions shown above.

We also checked the magnitude of the current density is OK!.

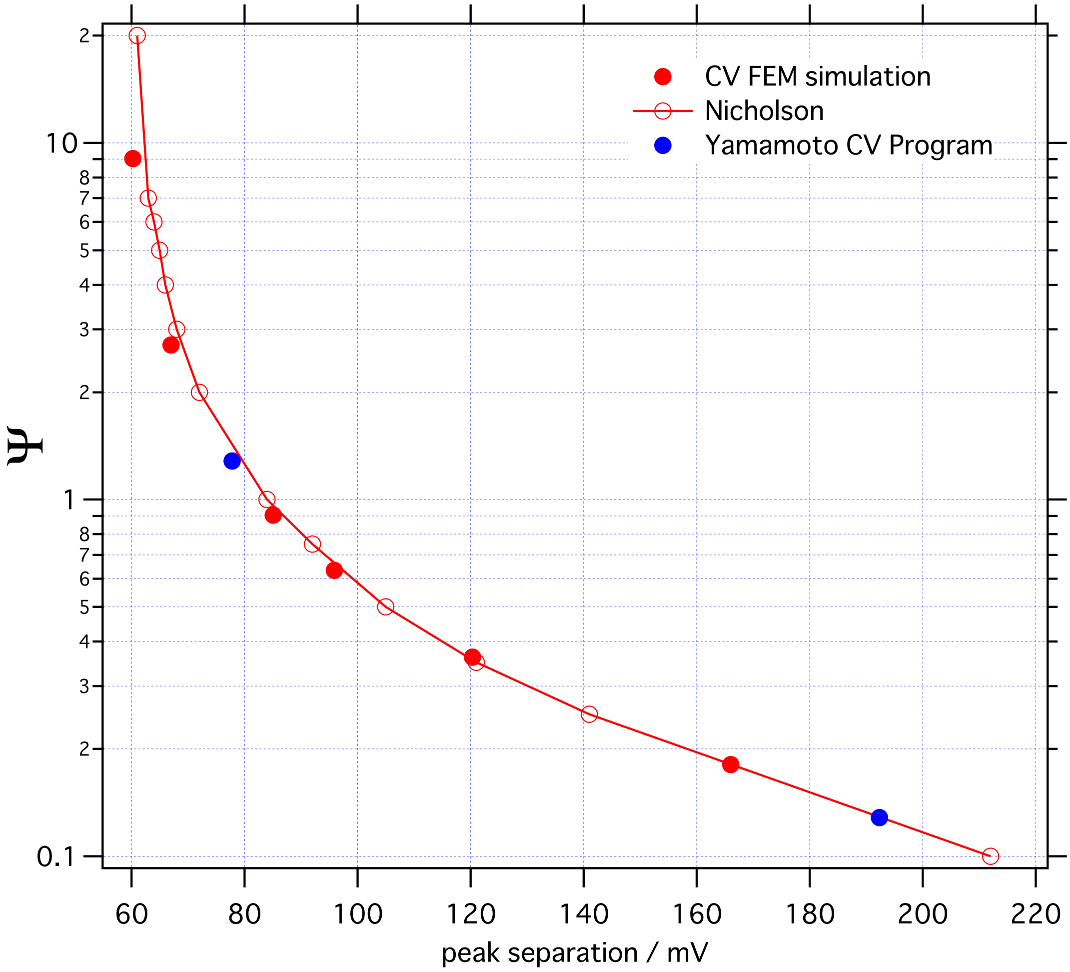

Figure 3: Nicholson plot

5 Adsorption (not finished)

Just consider the coverage $\theta_{\rm Ox}$ and $\theta_{\rm Rd}$ and the local equilibrium holds. The adsorption(desorption) equilibrium can be written in Langmuir, mean-filed, and quasi-chemical approximation. In the Langmuir approximation we obtain

$$k_{\rm ad}^{\rm Ox} c_{\rm Ox}(0,t)(1-\theta_{\rm Ox}) = k_{\rm de}^{\rm Ox}\theta_{\rm Ox}, \quad K^{\rm Ox} c_{\rm Ox}(0,t) = \frac{\theta_{\rm Ox}}{1-\theta_{\rm Ox}} \tag{61}$$

$$k_{\rm ad}^{\rm Rd} c_{\rm Rd}(0,t)(1-\theta_{\rm Rd}) = k_{\rm de}^{\rm Rd}\theta_{\rm Rd}, \quad K^{\rm Rd} c_{\rm Rd}(0,t) = \frac{\theta_{\rm Rd}}{1-\theta_{\rm Rd}} \tag{62}$$

If we assume that the redox reaction takes place at the adsorption site,

$$\frac{i}{nFA} = i_f + i_b = -k_f\theta_{\rm Ox}(t) + k_b\theta_{\rm Rd}(t) \tag{63}$$

$$\tag{64}$$

Which species can take part in the redox reaction?

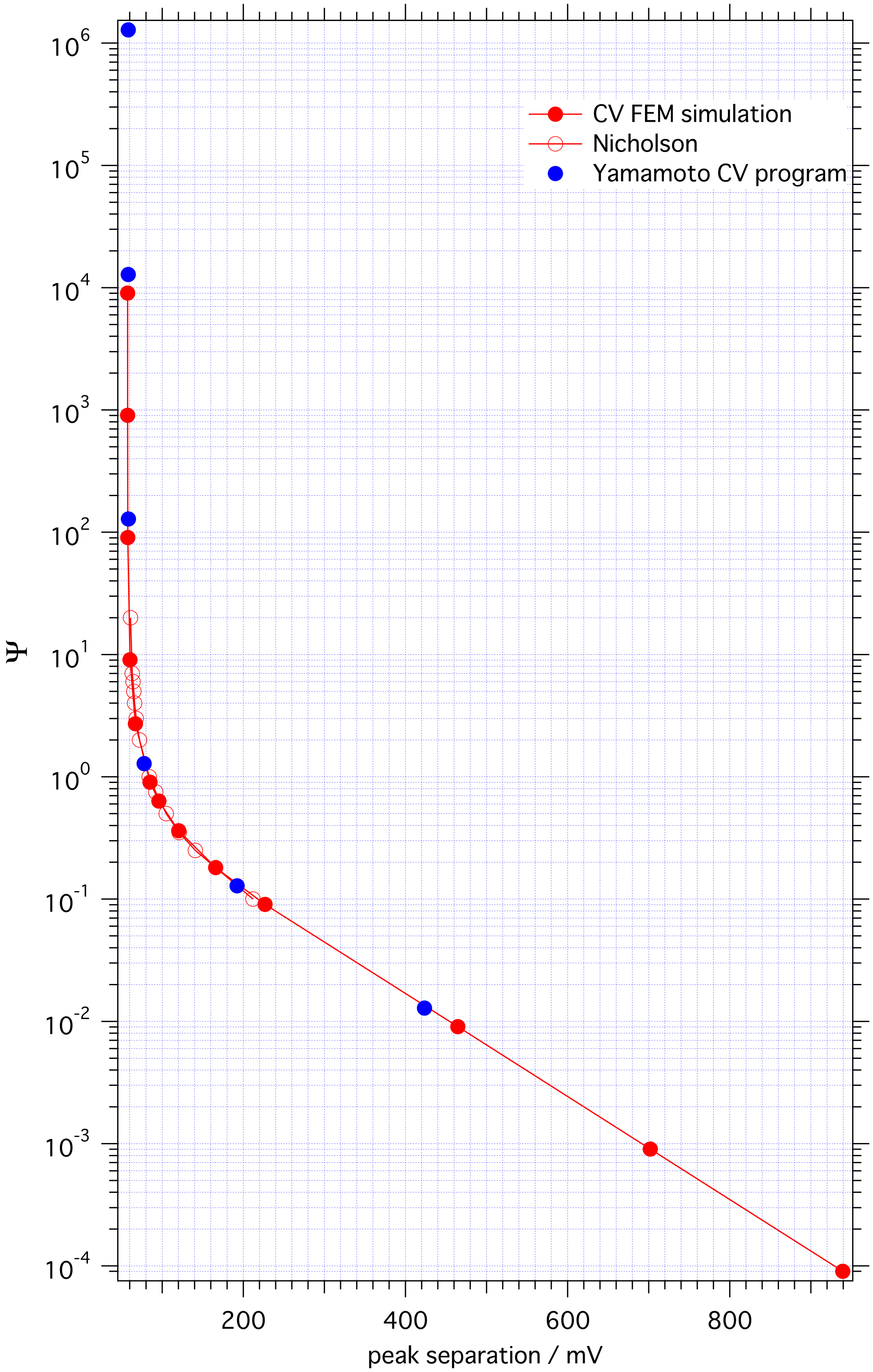

Figure 4: Nicholson plot

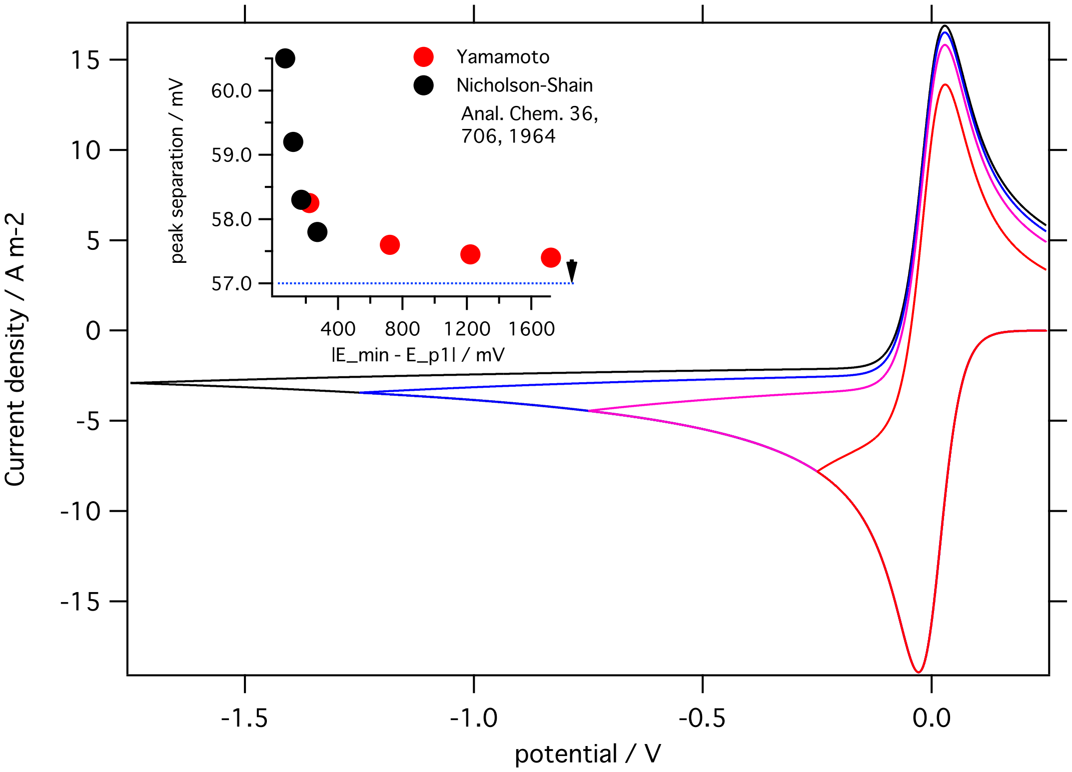

Figure 5: dependence of the peak separation on the switching potential for the reversible redox reaction. Simulation is done by the following condition. dox: diff. const. of Ox in m2 s-1 1.0E-009, drd: diff. const. of Rd in m2 s-1 1.0E-009, e0: start potential in V 0.25, v:sweep rate in V s-1; $v < 0$:redeuction -5.0E-002, eoxrd: formal potential of Ox-Red in V 0.0, imax: number of time mesh 20000, delta t: time step in s 1.0E-003, dm: step parameter usually taken as 0.45, delta z:mesh in z coor. in m 1.4907E-006, zmax: max in z / m 6.0E-004, kmax: number of mesh in z 403, cox0: 1000*c Ox in bulk / M 10.0, crd0: 1000*c Rd in bulk / M 0.0, ek0: rate const. k0 / m s-1 100, alpha:alpha parameter in Butlet-Volmer Eq. 0.5, temp:temp / K 300.0, en: n: Ox(z) + ne = Rd(z-n) 1.0.