

Electrografting of Mono-N-Boc-Ethylenediamine from an Acetonitrile/Aqueous NaHCO₃ Mixture

Hisham Hamzah¹, Guy Denuault¹, Philip Bartlett^{1*},
Aleksandra Pinczewska², Jeremy Kilburn³

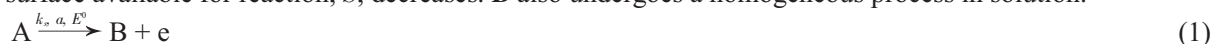
(1. Chemistry, University of Southampton, Southampton, SO17 1BJ, UK; 2. School of Biological and Chemical Sciences Queen Mary University of London, Mile End Road, London, E1 4NS, UK; 3. King's College, The University of Aberdeen, Aberdeen, AB24 3FX, UK)

Appendix I

Simulation of Cyclic Voltammograms for Inhibiting Adsorption

Here we follow the mechanism described by Bhugun and Savéant^[23].

In this scheme, the product of the electron transfer reaction, **B**, reacts with the electrode surface, **S**, to form a new species, **C**, which adsorbs and blocks further reaction. Hence, as the coverage of **C** increases, the amount of free surface available for reaction, **S**, decreases. **B** also undergoes a homogeneous process in solution.



Here, k_s , α , E^0 , k_a , and k_h are, respectively, the standard rate constant for electron transfer, transfer coefficient, standard potential, rate constant for adsorption and rate constant for the homogeneous reaction. The overall mechanism is mathematically shown below:

Partial Differential Equations

$$\frac{\partial c_A}{\partial t} = D_A \frac{\partial^2 c_A}{\partial x^2} \quad (4)$$

$$\frac{\partial c_B}{\partial t} = D_A \frac{\partial^2 c_B}{\partial x^2} - k_h c_B \quad (5)$$

where c_i and D_i are the concentration and diffusion coefficient of species **i**; x and t have their usual meaning.

Bhugun and Savéant assumed that the homogeneous reaction (3) is so fast that **B** has no time to diffuse in solution. Here we do not make this assumption and allow **B** to diffuse in solution.

Initial Conditions

$$c_A(x, 0) = c_A^b \quad (6)$$

$$c_B(x, 0) = c_B^b = 0 \text{ mol} \cdot \text{cm}^{-3} \quad (7)$$

$$n_C(0) = 0 \text{ mol} \quad (8)$$

$$S = S^0 = 1 \text{ cm}^2 \quad (9)$$

Superscript "b" indicates the bulk concentrations of the relevant species. n_C is the number of moles of **C** adsorbed, S is the amount of free surface and S^0 the whole surface available for adsorption.

These conditions correspond to the initial presence of **A** and absence of **B** from solution and to the availability of the whole electrode surface for adsorption of **C**.

Boundary Conditions at the Electrode $x = 0$

On the electrode, we assume that the flux of **A** is controlled by the potential through the Butler-Volmer expression. Bhugun and Savéant only included the rate expression for the reduction. In (10) we consider both reduction and oxidation processes.

$$D_A \frac{\partial c_A}{\partial x} \Big|_{x=0} = k_s \left(\exp \left((1-\alpha) \frac{F}{RT} (E-E^0) \right) c_A(0, t) - \exp \left(-\alpha \frac{F}{RT} (E-E^0) \right) c_B(0, t) \right) \quad (10)$$

For species **B** we account for diffusion and the surface reaction (2) as follows

$$D_A \frac{\partial c_A}{\partial x} \Big|_{x=0} = -D_B \frac{\partial c_B}{\partial x} \Big|_{x=0} + k_a c_B(0, t) \quad (11)$$

For species, **C**, we calculate the number of moles adsorbed, n_c , using

$$\frac{dn_c}{dt} = k_a c_B(0, t) S \quad (12)$$

with the amount of free surface, S , given by

$$S = S^0 - \frac{n_c}{\Gamma_{\max}} \quad (13)$$

where Γ_{\max} is the largest possible coverage of **C** on the electrode.

Boundary conditions for the bulk $x = \infty$

$$c_A(\infty, t) = c_A^b \quad (14)$$

$$c_B(\infty, t) = c_B^b = 0 \text{ mol} \cdot \text{cm}^{-3} \quad (15)$$

The bulk of the solution is not affected by the electrode reactions and the concentrations of **A** and **B** recover their initial values.

Calculation of the Current

In (16) we calculate the current from the flux of **A** towards the electrode but account for the change in available electrode area through the value of S .

$$i = FSD_A \frac{\partial c_A}{\partial x} \Big|_{x=0} \quad (16)$$

The current could also be estimated using equation (17).

$$i = FS k_s \left(\exp \left((1-\alpha) \frac{F}{RT} (E-E^0) \right) c_A(0, t) - \exp \left(-\alpha \frac{F}{RT} (E-E^0) \right) c_B(0, t) \right) \quad (17)$$

Finite Difference Scheme

There are many suitable numerical schemes to solve the above equations. Here we use a very simple finite difference approach to compute the concentrations of **A** and **B** in solution, coverage of **C** and available surface area S .

Expressing (4) and (5) in terms of finite differences and rearranging to obtain estimates for the concentrations of **A** and **B** we have

$$c_{A_i}^{k+1} = \frac{D_A \Delta t}{\Delta x} \left(c_{A_{i+1}}^k - 2c_{A_i}^k + c_{A_{i-1}}^k \right) \quad (18)$$

$$c_{B_i}^{k+1} = \frac{D_B \Delta t}{\Delta x} \left(c_{B_{i+1}}^k - 2c_{B_i}^k + c_{B_{i-1}}^k \right) - k_a c_{B_i}^k \quad (19)$$

where $c_{A_i}^k$ and $c_{B_i}^k$ are respectively the concentrations of **A** and **B** at time index $k+1$ and space index i .

From (10) and (11) we have

$$D_A \frac{c_{A_2}^k - c_{A_1}^k}{\Delta x} = k_s \left(\exp \left((1-\alpha) \frac{F}{RT} (E-E^0) \right) c_{A_1}^k - \exp \left(-\alpha \frac{F}{RT} (E-E^0) \right) c_{B_1}^k \right) \quad (20)$$

$$D_A \frac{c_{A_2}^k - c_{A_1}^k}{\Delta x} = -D_B \frac{c_{B_2}^k - c_{B_1}^k}{\Delta x} + k_a c_{B_1}^k \quad (21)$$

After some rearrangements, see below, (20) and (21) lead to a system of two simultaneous equations with only $c_{A_1}^k$ and $c_{B_1}^k$ as unknowns.

$$\text{Let } \varepsilon = \frac{F}{RT}(E - E^0) \quad (22)$$

Rearranging (21) yields

$$-D_A c_{A_1}^k - (D_B + \Delta x k_a) c_{B_1}^k + D_A c_{A_2}^k + D_B c_{B_2}^k = 0 \quad (23)$$

From (20) we get

$$D_A c_{A_2}^k - D_A c_{A_1}^k = \Delta x k_s (\exp((1-\alpha)\varepsilon) c_{A_1}^k - \exp(-\alpha\varepsilon) c_{B_1}^k) \quad (24)$$

i.e.

$$-(D_A + \Delta x k_s \exp(1-\alpha)\varepsilon) c_{A_1}^k + \Delta x k_s \exp(-\alpha\varepsilon) c_{B_1}^k + D_A c_{A_2}^k = 0 \quad (25)$$

We rewrite (23) and (25) as follows

$$a c_{A_1}^k + b c_{B_1}^k + c = 0 \quad (26)$$

$$d c_{A_1}^k + e c_{B_1}^k + f = 0 \quad (27)$$

with

$$a = -D_A \quad (28)$$

$$b = -(D_B + \Delta x k_a) \quad (29)$$

$$c = D_A c_{A_2}^k + D_B c_{B_2}^k \quad (30)$$

$$d = -(D_A + \Delta x k_s \exp((1-\alpha)\varepsilon)) \quad (31)$$

$$e = \Delta x k_s \exp(-\alpha\varepsilon) \quad (32)$$

$$f = D_A c_{A_2}^k \quad (33)$$

From (26)

$$c_{A_1}^k = -\frac{b c_{B_1}^k + c}{a} \quad (34)$$

Substituting for $c_{A_1}^k$ in (27) gives

$$d \left(-\frac{b c_{B_1}^k + c}{a} \right) + e c_{B_1}^k + f = 0 \quad (35)$$

i.e.

$$c_{B_1}^k = \left(\begin{array}{c} -f + \frac{cd}{a} \\ e - \frac{bd}{a} \end{array} \right) \quad (36)$$

Together (34) and (36) provide the surface concentrations for **A** and **B** for every time step in the finite difference algorithm.

Applying the finite difference approximation to (12) we get

$$n_C^{k+1} = n_C^k + \Delta t k_a c_{B_1}^k S^k \quad (37)$$

while for (13) we have

$$S^k = S^0 - \frac{n_C^k}{\Gamma_{\max i}} \quad (38)$$

and for (16)

$$i = FS^k D_A \frac{c_{A_2}^k - c_{A_1}^k}{\Delta x} \quad (39)$$

Table of Symbols

Symbol	Unit	Description
α		transfer coefficient
c_A	$\text{mol} \cdot \text{cm}^{-3}$	concentration of A
c_A^b	$\text{mol} \cdot \text{cm}^{-3}$	bulk concentration of A
c_B	$\text{mol} \cdot \text{cm}^{-3}$	concentration of B
c_B^b	$\text{mol} \cdot \text{cm}^{-3}$	bulk concentration of B
D_A	$\text{cm}^2 \cdot \text{s}^{-1}$	diffusion coefficient of A
D_B	$\text{cm}^2 \cdot \text{s}^{-1}$	diffusion coefficient of B
E	V	electrode potential
E^0	V	standard potential
ε		dimensionless potential
Γ_{max}	$\text{mol} \cdot \text{cm}^{-2}$	largest possible coverage of C on the electrode ($1/\Gamma_{\text{max}}$ corresponds to the footprint of 1 mole of C)
i	A	current
k_a	$\text{cm} \cdot \text{s}^{-1}$	rate constant for adsorption
k_h	s^{-1}	rate constant for homogeneous reaction
k_s	$\text{cm} \cdot \text{s}^{-1}$	standard rate constant for electron transfer
n_C	mol	moles of C on electrode
S^0	cm^2	electrode surface available for reaction at $t = 0$
S	cm^2	electrode surface available for reaction at t
t	s	time
x	cm	distance normal to electrode

Matlab Codes

```

%
% This program simulates repeated cyclic voltammograms for the mechanism
%
% A->B+e, ks, alfa, E0
% B+S->C, ka, S free surface, C sticks to the ede and inhibits further rxn
% B->P, kh, homogeneous process
%
% As the coverage of C increases the amount of free surface decreases
%
clear
cAb = 1e-6; % / mol cm-3, bulk cntn of A
cBb = 0; % / mol cm-3, bulk cntn of B
DA = 1e-5; % / cm2 s-1, diffusion coefficient of A
DB = DA; % / cm2 s-1, diffusion coefficient of B
GCmax = 1e9; % / mol cm-2, largest coverage of C
S0 = 1; % / cm2; free ede area available at t=0
ka = 0.01; % / cm s-1, rate cst for adsorption of B

```

```

kh = 0.1; % / s-1, rate constant for decomposition of B in solution
ks = 1; % / cm s-1, standard rate cst for e- transfer
alfa = 0.5; % transfer coefficient
Ei = -0.5; % / V, start of ptl sweep
Ef = +0.5; % / V, end of ptl sweep
E0 = 0; % / V, standard ptl
nu = 0.1; % / V s-1, sweep rate
F = 96485; % / C mol-1
R = 8.314; % / J K-1 mol-1
T = 298; % / K
lambda = 0.45; % used to ensure FD algorithm remains stable
ncycles = 4; % number of voltammetric cycles
kmax = 50; % number of iterations over forward scan of CV
npinrl = 1;
while npinrl < 2 % ensures at least two points within reaction layer
    dt = abs(Ef-Ei)/nu/kmax; % / s, time increment
    dx = sqrt(DA*dt/lambda); % / cm, space increment
    npinrl = sqrt(DB/kh)/dx; % number of points within reaction layer
    kmax = 2*kmax;
end
kmax2 = 2 * kmax; % number of iterations over forward and reverse scans
kmaxn = kmax2 * ncycles; % number of iterations for all the cycles
imax = fix(6 * sqrt(kmaxn * lambda)); % converts imax to an integer
cA = cAb * ones(imax,kmaxn); % / mol cm-3, cntn of A
cB = cBb * ones(imax,kmaxn); % / mol cm-3, cntn of B
Cur = zeros(kmaxn,1); % / A, current
Pot = zeros(kmaxn,1); % / V, potential
Time = zeros(kmaxn,1); % / s, time
nC = zeros(kmaxn,1); % / mol, moles of C on ede
dE = (Ef - Ei) / kmax; % / V, potential increment
E = Ei; % initialises potential
for k=1:kmaxn-1 % start of time loop
    epsi = F/R/T*(E-E0);
    a = -DA; % a,b,c,d,e,f are the coefficients of 2 simultaneous eq. to
    b = -(DB+dx*ka); % solve for cA(1,k) and cB(1,k)
    c = DA*cA(2,k)+DB*cB(2,k);
    d = -(DA+dx*ks*exp((1-alfa)*epsi));
    e = dx*ks*exp(-alfa*epsi);
    f = DA*cA(2,k);
    cB(1,k) = (-f+c*d/a)/(e-b*d/a); % electrode boundary condition
    cA(1,k) = -(c+b*cB(1,k))/a; % electrode boundary condition
    Sfree = S0 - nC(k) / GCmax; % surface not covered by C
    Cur(k) = F*Sfree*DA*(cA(2,k)-cA(1,k))/dx;
    Time(k) = k * dt;
    Pot(k) = E;
    nC(k+1) = nC(k) + dt * ka * cB(1,k) * Sfree;

```

```

for i = 2 : imax - 1 % diffusion loop
    cA(i,k+1) = cA(i,k) + lambda * ...
        (cA(i+1,k) - 2 * cA(i,k) + cA(i-1,k));
    cB(i,k+1) = cB(i,k) + lambda * ...
        (cB(i+1,k) - 2 * cB(i,k) + cB(i-1,k)) - dt * kh * cB(i,k);
end
E = E + dE;
if E > Ef
    dE = -dE; %changes sign of ptl increment at end of scans
end
if E < Ei
    dE = -dE;
end
end
%plot results
x = ones(imax,1);
for i = 1 : imax
    x(i) = i*dx;
end
Emin = min(Ei,Ef);
Emax = max(Ei,Ef);
for k = 1:2:kmaxn-1 % this loops animates the plots
    subplot(4,1,1) % prepares plot1 in a 4x1 array of plots on the page
    plot(Time(1:k), Pot(1:k));
    xlabel('Time / s');ylabel('Pot / V');
    xlim([0 max(Time)]);ylim([1.1*Emin 1.1*Emax]);
    subplot(4,1,2); % prepares plot2 in a 4x1 array of plots on the page
    plot(Time(1:k), Cur(1:k));
    xlabel('Time / s');ylabel('Cur / A');
    xlim([0 max(Time)]);ylim([1.1*min(Cur) 1.1*max(Cur)]);
    subplot(4,1,3); % prepares plot3 in a 4x1 array of plots on the page
    plot(Pot(1:k), Cur(1:k));
    xlabel('Pot / V');ylabel('Cur / A');
    xlim([Emin Emax]);ylim([1.1*min(Cur) 1.1*max(Cur)]);
    subplot(4,1,4); % prepares plot4 in a 4x1 array of plots on the page
    plot(x,cA(:,k),x,cB(:,k));
    ylim([0 1.1*cAb]);xlabel('x / cm');ylabel('c(x,t) / mol cm-3');
    pause(0.01);
end
% Ouput results to spreadsheet
FileOut = 'FD_CV_inhibition_A2B_BpS2C_B2P';
XlsSheet = 'FD_Sim';
xlswrite(FileOut,{'E / V'},XlsSheet,'A11')
xlswrite(FileOut,{'i / A'},XlsSheet,'B11')
xlswrite(FileOut,Pot(1:kmaxn-1),XlsSheet,'A12');
xlswrite(FileOut,Cur(1:kmaxn-1),XlsSheet,'B12');

```