Numerical Modeling of Electrochemical Engineering

Version 1.0

Yuanchao Li, PhD

Email:liyuanchao616@gmail.com

July 2019

Contents

1 Preface **2**

2 Single electrode model **2**

2.1 Chronoamperometry 3

2.2 Linear sweep voltammetry 14

3 Hydrogen-Oxygen PEM fuel cell modeling **18**

3.1 LU decomposition of linear system 18

3.2 Physical model of PEM fuel cell 18

3.3 Numerical solution of PEM fuel cell 18

1 Preface

Thousands of fresh graduate student start their research in the field of electrochemistry every year, ranging from the electrocatalyst to the lithium battery. They may carefully place three electrodes in the electrolyte and set up the parameters like the initial voltage, cut-off voltage, and scan rate based on the supervision from their senior lab mates. They will be excited to see the duck-shaped cyclic voltammograms. Then they pour out the information like the peak of oxidation and reduction in a report or a presentation of a group meeting. However, they may be soon tired of doing the CV again and again. They start to read the book like *Electrochemical Systems* and try to figure out the mechanism behind the duck-shaped CVs. Some of them quickly run through the maze made of equations and arrive at the destination. But most of the students are frustrated in the maze due to the lack of knowledge of calculus, numerical solution, or the skills to write the code. They tried to seek help but received rare responses. Though I am from the academic tree of John Newman, I have to admit that I was once in the team of slow movers. Luckily I didn't give up and slowly walked through the maze. After getting a Ph.D., I hope to contribute to the community by sharing my knowledge. I will establish two models in this file, the model of a single electrode and the model of a whole fuel cell. In the version 1.0, the single electrode model was described. The model of a whole fuel cell model will be added in the version 2.0. If I am still have some times, I'd like to update a lithium battery model in the version 3.0. But it cannot be guaranteed. I will present the physical model, mathematical knowledge related to solving the model, and the source code written by Python. And I tried to avoid the highly efficient but implicit Python code. All steps in the derivation will be included.

2 Single electrode model

A reversible reaction, $O + ne^- \longleftrightarrow R$, occurs on the working electrode. A scheme of working electrode is shown in Figure 2.1.

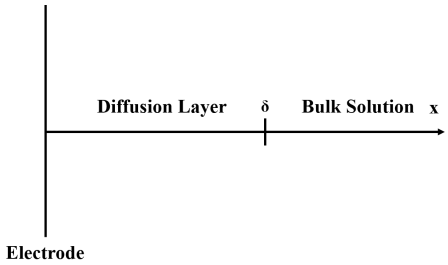

Figure 2.1: Scheme of single electrode {fig:scheme of

Every numerical modeling has their own assumptions. Here, several assumptions for this modeling have to be introduced. We can assume that the solution starts with species O initially and has no R. Then the initial conditions can be written as,

$$C_O(x,0) = C_O^* \tag{2.1}$$ {eq1}

and

$$C_R(x,0) = 0 \tag{2.2}$$ {eq2}

Another assumption is the semi-infinite boundary conditions. One can normally assume that at large distances from the electrode, the concentration reaches to a constant value since the electrolysis cell is

usually large compared to the length of diffusion.

$$\lim_{x\to\infty} C_O(x,t) = C_O^* \tag{2.3}$$ {eq3}

and

$$\lim_{x\to\infty} C_R(x,t) = 0 \tag{2.4}$$ {eq4}

We need another boundary condition on the electrode surface. It usually depends on the assumption of the kinetics of the electrochemical reaction. The general current-potential characteristic like the Butler-Volmer equation or its special case, Nernst equation, can be used. To simplify our instruction, I'd like to assume the electrochemical reaction as the highly reversible one, which is called the Nernstian electrode process. Its electrode kinetics is so fast that the electrode potential and the surface concentrations are in local Nernstian balance at all times regardless of the details of the mechanism and current flow.

$$E = E^{0'} + \frac{RT}{nF} ln \frac{C_O(0,t)}{C_R(0,t)} \tag{2.5}$$ {eq5}

With those assumptions and boundary conditions, we can start to think about the electrochemistry model on a single working electrode.

2.1 Chronoamperometry

The first case I'd like to talk about is the chronoamperometry experiment, which means that the potential was held as a constant for a long time. The i-E curve and concentration profile will be modeled. A detailed model and its derivation of cathodic (reduction) chronoamperometry will be demonstrated.

When the applied potential is lower than the standard reduction potential, there is a cathodic (reduction) reaction on the working electrode. If the applied potential is higher than the standard reduction potential, there is an anodic (oxidation) reaction on the working electrode. IUPAC defines anodic current as positive while the cathodic current as negative. Taking cathodic reaction as an example, the flux of O should transports toward the electrode surface (negative x direction, negative N_O value) and species R transports away from the electrode surface to the bulk solution (positive x direction, positive N_R) since O is reduced to R in the cathodic reaction.

$$\frac{i}{A} = N_O nF \tag{2.6}$$ {eq6}

$$\frac{i}{A} = -N_R nF \tag{2.7}$$ {eq7}

, where i is the current, n is the stoichiometry of electron in the reaction, A is the electrode surface contacting with electrolyte, and F is the Faradic constant equaling to 96485 $\mathrm{C/mol}$. Equation 2.6 and 2.7 reveal the relationship between the current and mass transfer in electrolyte. The unit of current is ampere (A) which can be written as $\mathrm{C/s}$. 1 A indicates 1 C electrons is consumed on the electron in a second. The Faradic constant points out that 1 mole of electrons contains 96485 C charges. By the electrochemical reaction formula ($\mathrm{O + ne^- \longleftrightarrow R}$), every mole of consumed/generated electrons corresponds to $\frac{1}{n}$ mole of O and R respectively. Since the molar flux is the rate of molar flow per unit area, the current is also converted into current density by dividing an electrode surface area. Equation 2.6 and 2.7 answer a simple but important question, how many moles of reactant/product are needed to be transported to/away the electrode surface in the unit time and area for the generation of a certain current density? A more significant physical meaning is that the bridge between the measurable parameter (current,i) and the microparameters (N_O, N_R) has been established.

Nernst-Planck equation describes the molar flux in the electrolyte,

$$N_i(x) = -D_i\frac{\partial C_i}{\partial x} - \frac{z_i F}{RT}D_i C_i\frac{\partial \phi}{\partial x} + C_i v(x) \tag{2.8}$$ {eq8}

where the molar flux $N_i(x)$ is written as the sum of diffusion, migration, and convection. In the chronoamperametry test, the voltage is held as a constant while current changes vs. time. Based on Equation 2.6 and 2.7, then the molar fluxes of O and R are also expected to change vs. time. The time-dependent Nernst-Plank equation has to be introduced.

$$\frac{\partial C_i}{\partial t} = -\nabla N_i \tag{2.9}$$ {eq9}

It has to be pointed out that Equation 2.9 is derived from the mass conservation. In the more general form, a homogeneous reaction rate of species i,R_i, should be added to the right side of Equation 2.9. Most of the electrochemical reactions are heterogeneous reactions which only happen on the interface of electrode and electrolyte. Then the R_i is zero. There is one special case. In the homogeneous model of a porous electrode, the porous electrode like the gas diffusion electrode (GDE) of fuel cell including the electrolyte inside the pores is viewed as a homogeneous material. In this case, the reaction on the solid surface can be viewed as the homogeneous reaction inside the porous electrode.

Let us go back to Equation 2.8. There are three terms in the molar flux, diffusion, migration, and convection. If the solution is not stirred, the convection is zero. In most of three-electrode electrochemical experiments, especially with supporting electrolyte (like sulfuric acid or alkaline solution), the potential difference ($\nabla\phi$) is negligible. So in this case, only diffusion will be considered for the mass transfer. Then we can rewrite the time-dependent Nernst-Plank equation, Equation 2.9.

$$\frac{\partial C_i(x,t)}{\partial t} = D_i\frac{\partial^2 C_i(x,t)}{\partial x^2} \tag{2.10}$$ {eq10}

The boundary conditions for Equation 2.10 are Equation 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, and 2.7. The boundary conditions from Equation 2.5, 2.6, and 2.7 may be a little confused. We will explain it in the process of derivation.

To solve the time-dependent Equation 2.10, Laplace transform will be used. It transforms a function of a real variable t (often time) to a function of a complex variable s (complex frequency). In the application of solving differential equations, the Laplace transform reduces a linear differential equation to an algebraic equation, which can then be solved by the formal rules of algebra.

Next, we'd like to introduce how to use Laplace transform to solve the differential equation. The basic form of Laplace transformation is,

$$\mathscr{L}\{F(t)\} = \int_0^\infty e^{-s(t)}F(t)dt \tag{2.11}$$

where the $\mathscr{L}\{F(t)\}$ can also be written as f(s).

In the transform of partial equations,

$$\mathscr{L}\{\frac{dF(t)}{dt}\} = sf(s) - F(0) \tag{2.12}$$ {eq11}

The general form is,

$$\mathscr{L}\{F^{(n)}\} = s^n f(s) - s^{n-1}F(0) - s^{n-2}F'(0)...F^{n-1}(0) \tag{2.13}$$

Figure 2.2 shows the Laplace transform on the common functions. It can be used for the Laplace transform and inverse Laplace transform. For example, $\mathscr{L}\{t\} = 1/s^2$. On the other hand, we know $F(t) = t$ if we see $\mathscr{L}^{-1}\{t\} = 1/s^2$.

$F(t)$	$f(s)$
A(constant)	A/s
e^{-at}	$1/(s+a)$
$\sin at$	$a/(s^2+a^2)$
$\cos at$	$s/(s^2+a^2)$
$\sinh at$	$a/(s^2-a^2)$
$\cosh at$	$s/(s^2-a^2)$
t	$1/s^2$
$t^{(n-1)}/(n-1)!$	$1/s^n$
$(\pi t)^{-1/2}$	$1/s^{1/2}$
$2(t/\pi)^{1/2}$	$1/s^{3/2}$
$\frac{x}{2(\pi k t^3)^{1/2}}\,[\exp(-x^2/4kt)]$	$e^{-\beta x}$, where $\beta = (s/k)^{1/2}$
$\left(\frac{k}{\pi t}\right)^{1/2}\,[\exp(-x^2/4kt)]$	$e^{-\beta x}/\beta$
$\operatorname{erfc}[x/2(kt)^{1/2}]$	$e^{-\beta x}/s$
$2\left(\frac{kt}{\pi}\right)^{1/2}\exp(-x^2/4kt) - x\operatorname{erfc}[x/2(kt)^{1/2}]$	$e^{-\beta x}/s\beta$
$\exp(a^2t)\operatorname{erfc}(at^{1/2})$	$\frac{1}{s^{1/2}(s^{1/2}+a)}$

Figure 2.2: Laplace Transforms of Common Functions {fig:Laplace tr

By using Equation 2.12 on 2.10, one can get,

$$s\overline{C_O}(x,s) - \overline{C_O}(x,0) = D_i\frac{\partial^2\overline{C_O}(x,s)}{\partial x^2} \tag{2.14}$$ {eq12}

$C_O(x,0) = C_O^*$ can be transformed by Laplace transform as the first row in Figure 2.2, $\mathscr{L}\{C^*\} = C_O^*/s$. After we plug it in Equation 2.14, one can get,

$$\frac{\mathrm{d}\overline{C_O}(x,s)}{\mathrm{d}x} - \frac{s}{D_O}\overline{C_O}(x,s) = -\frac{C^*}{D_O} \tag{2.15}$$ {eq13}

We can see $\frac{\partial C_i}{\partial t}$ has been converted into an algebra form with s instead of t. Equation 2.15 is very easy to be solved as,

$$\overline{C_O}(x,s) = \frac{C_O^*}{s} + A(s)exp(-\sqrt{\frac{s}{D_O}}x) + B(s)exp(\sqrt{\frac{s}{D_O}}x) \tag{2.16}$$ {eq14}

From the boundary condition, $\lim_{x\to\infty} C_O(x,t) = C_O^*$, we can know $\lim_{x\to\infty}\overline{C_O}(x,s) = C_O^*/s$. Then $B(s)$ is zero in Equation 2.16. Equation 2.16 can be rewritten as,

$$\overline{C_O}(x,s) = \frac{C_O^*}{s} + A(s)exp(-\sqrt{\frac{s}{D_O}}x) \tag{2.17}$$ {eq15}

Similarly, the concentration profile of R after Laplace transform is,

$$\overline{C_R}(x,s) = B(s)exp(-\sqrt{\frac{s}{D_R}}x) \tag{2.18}$$ {eq16}

Here, it's worth being pointed out that the derivation of Equation **??** and **??** is independent of the assumption of highly reversible reaction (Nernstian equation, Equation 2.5). So these two equations can be viewed as a general form of concentration profile in Laplace transform.

By now, the boundary conditions from Equation 2.1, 2.2, 2.3, and 2.4 have been used to get Equation 2.17 and 2.18. Next, we have to use Equation 2.6 and 2.7 to connect Equation 2.17 and 2.18. Since the electrochemical reaction happen on the electrode surface ($x = 0$), we can get information about $\overline{C_O}(0,s)$ and $\overline{C_R}(0,s)$.

$$\begin{aligned}\frac{\bar{i}(s)}{nFA} &= -D_O\frac{\mathrm{d}\overline{C_O}(x,s)}{\mathrm{d}x}|_{x=0} = -D_O(-A(s)\sqrt{\frac{s}{D_O}})\\ \frac{\bar{i}(s)}{nFA} &= D_R\frac{\mathrm{d}\overline{C(x,s)}}{\mathrm{d}x}|_{x=0} = D_R(-B(s)\sqrt{\frac{s}{D_R}})\end{aligned} \tag{2.19}$$ {eq17}

From Equation 2.19, we can easily get,

$$\begin{aligned}B(s) &= -\sqrt{\frac{D_O}{D_R}}A(s)\\ &= -\epsilon A(s)\end{aligned} \tag{2.20}$$ {eq18}

The last boundary condition, Equation 2.5, needs to be used as solve $A(s)$. Equation 2.5 assumes that the electrochemical reaction is very fast and highly reversible. It describes the relationship of the

applied potential E and the surface concentration of O/R at t. In the chronoamperometry test, the applied potential is a known number. With the combination of Equation 2.5, 2.17, 2.18, and 2.20, one gets,

$$A(s) = -\frac{C_O^*/s}{1+\epsilon\theta}, \text{where } \theta = exp[\frac{RT}{nF}(E - E^{0'})] \tag{2.21}$$ {eq18-s}

Now, the concentration profiles of R and O with Laplace transform have been solved as,

$$\begin{aligned} \overline{C_O}(x,s) &= \frac{C_O^*}{s} - \frac{C_O^*/s}{1+\epsilon\theta} exp(-\sqrt{\frac{s}{D_O}}x) \\ \overline{C_R}(x,s) &= \frac{\epsilon C_O^*/s}{1+\epsilon\theta} exp(-\sqrt{\frac{s}{D_R}}x) \end{aligned} \tag{2.22}$$ {eq19}

By checking the 13th row in Figure 2.2, we can do the inverse Laplace transform on Equation 2.22 to get,

$$\begin{aligned} C_O(x,t) &= C_O^* - \frac{C_O^*}{1+\epsilon\theta}\,\mathrm{erfc}\,(\frac{x}{2\sqrt{D_O t}}) \\ C_R(x,t) &= \frac{\epsilon C_O^*}{1+\epsilon\theta}\,\mathrm{erfc}\,(\frac{x}{2\sqrt{D_R t}}) \end{aligned} \tag{2.23}$$ {eq20}

The erfc is the complementary error function, $1-\mathrm{erf}\,(x) = 1 - 2/\sqrt{\pi}\int_0^x e^{-t^2}dt$.Similarly, the 9th row in Figure 2.2 can be applied on the combination of Equation 2.19 and 2.21 to get

$$i(t) = -\frac{C_O^* nFA}{1+\epsilon\theta}\sqrt{\frac{D_O}{\pi t}} \tag{2.24}$$ {eq21}

By now, the analytic solutions of concentration profiles of R/O and i-E curve have been obtained. We just need to write a Python code to solve and plot them. The Python code used to solve this modeling is attached here.

```
import matplotlib.pyplot as plt
import numpy as np
from scipy import special

C_OB = 1 #mol/m3
C_RB = 1 #mol/m3
D_O = 1E-9 #m2/s
D_R = 1E-9 #m2/s
F = 96485 #C/mol
R = 8.314 #J / mol. K
pi = 3.14
A = 1E-4 #1m2
T = 298 #K
epsilon = np.sqrt(D_O / D_R)
color=['black','red','blue','purple','c','grey','green']
C_Rt = []
C_Ot = []
C_R = []
C_O = []
```

```
t_i = []
i_a = []
i_c = []
i_ai = []
i_ci = []


def CurrentVsTime(E_ai,E_ci,E_0,n):
#applied anodic voltage, appllied cathodic voltage, equilibrium voltage,
#number of electrons
  E_a = []
  E_c = []
  t = np.linspace(0.001, 1, 100)
  E_a.append(E_ai)
  E_c.append(E_ci)
  while np.abs(E_ai-E_0) < 0.1:
    theta_a = np.exp(n * F * (E_ai - E_0) / R / T)
    theta_c = np.exp(n * F * (E_ci - E_0) / R / T)
    i_ai = C_RB * n * F *A * np.sqrt(D_R / pi / t) *theta_a * epsilon / \
    (1 + theta_a * epsilon)
    i_ci = - C_OB * n * F *A * np.sqrt(D_O / pi / t) / \
    (1 + theta_c * epsilon)
    i_a.append(i_ai)
    i_c.append(i_ci)
    E_ai = E_ai + 0.02
    E_ci = E_ci - 0.02
    E_a.append(E_ai)
    E_c.append(E_ci)

  plt.figure()

  for i in range (0,len(E_a)-1):
    overpotential = E_a[i] - E_0
    plt.plot(t,i_a[i],'s-',color = color[i],\
      label='Anodic current at overpotential of %.3fV' \
      % overpotential,markevery=50)
  for i in range (0,len(E_c)-1):
    overpotential = E_c[i] - E_0
    plt.plot(t,i_c[i],'*-',color = color[i],\
      label='Cathodic current at overpotential of %.3fV' \
      % overpotential,markevery=50)
  del E_c[:]
  del E_a[:]
  ax = plt.gca()
  ax.spines['right'].set_color('none')
  ax.spines['top'].set_color('none')
  ax.xaxis.set_ticks_position('bottom')
  ax.spines['right'].set_color('none')
  ax.spines['top'].set_color('none')
```

```
    ax.xaxis.set_ticks_position('bottom')
    ax.yaxis.set_ticks_position('left')
    ax.spines['bottom'].set_position(('data',0))
    ax.spines['left'].set_position(('data',0))
    plt.ylabel(r'Current_A')
    plt.xlabel(r'time_s')
    plt.legend(loc='best',edgecolor='none')
    plt.savefig('current_vs_time.png',dpi=300)
    plt.show()

def ConcentrationProfileReduction(E_c,E_0,n):
#appllied cathodic voltage, equilibrium voltage, number of electrons
    x = np.linspace(0,2E-4,1000)
    theta_c = np.exp(n * F * (E_c - E_0) / R / T)
    t = 0.001
    t_i.append(t)
    while t < 11:
      C_Ot = C_OB - C_OB * special.erfc(x / 2 / np.sqrt(D_O * t)) / \
      (1 + epsilon * theta_c)
      C_Rt = C_OB * epsilon * special.erfc(x / 2 / np.sqrt(D_O * t)) / \
      (1 + epsilon * theta_c)
      C_R.append(C_Rt)
      C_O.append(C_Ot)
      t = t*10
      t_i.append(t)
    plt.figure()
    for i in range(0,len(t_i)-1):
      plt.plot(x,C_O[i],'o-',color = color[i],\
        label='t=%.3fs,C_O(x,t)' % t_i[i],markevery=50)
    for i in range(0,len(t_i)-1):
      plt.plot(x,C_R[i],'*-',color = color[i],\
        label='t=%.3fs,C_R(x,t)' % t_i[i],markevery=50)
    del C_R[:]
    del C_O[:]
    del t_i[:]
    ax = plt.gca()
    ax.spines['right'].set_color('none')
    ax.spines['top'].set_color('none')
    ax.xaxis.set_ticks_position('bottom')
    ax.spines['right'].set_color('none')
    ax.spines['top'].set_color('none')
    ax.xaxis.set_ticks_position('bottom')
    ax.yaxis.set_ticks_position('left')
    ax.spines['bottom'].set_position(('data',0))
    ax.spines['left'].set_position(('data',0))
    plt.ticklabel_format(style='sci', axis='x', scilimits=(0,0))
    plt.ylabel(r'Concentraion_$mol/m^3$')
    plt.xlabel(r'x_m')
    plt.legend(loc='best',edgecolor='none')
```

```
  plt.title("Cathodic voltage:%.2fV; Equilibrium potential:%.2fV" % (E_c,E_0))
  plt.savefig('Concentration_profile_reduction.png',dpi=300)

def ConcentrationProfileOxidation(E_a,E_0,n):
#applied anodic voltage, equilibrium voltage, number of electrons
  x = np.linspace(0,2E-4,1000)
  theta_a = np.exp(n * F * (E_a - E_0) / R / T)
  t = 0.001
  t_i.append(t)
  while t < 11:
    C_Rt = C_RB - C_RB * theta_a * epsilon * special.erfc(x / 2 / \
      np.sqrt(D_R * t)) / (1 + epsilon * theta_a)
    C_Ot = C_RB * theta_a * special.erfc(x / 2 / \
      np.sqrt(D_R * t)) / (1 + epsilon * theta_a)
    C_R.append(C_Rt)
    C_O.append(C_Ot)
    t = t*10
    t_i.append(t)
  plt.figure()
  for i in range(0,len(t_i)-1):
    plt.plot(x,C_O[i],'o-',color = color[i],\
      label='t=%.3fs,C_O(x,t)' % t_i[i],markevery=50)
  for i in range(0,len(t_i)-1):
    plt.plot(x,C_R[i],'*-',color = color[i],\
      label='t=%.3fs,C_R(x,t)' % t_i[i],markevery=50)
  del C_R[:]
  del C_O[:]
  del t_i[:]
  ax = plt.gca()
  ax.spines['right'].set_color('none')
  ax.spines['top'].set_color('none')
  ax.xaxis.set_ticks_position('bottom')
  ax.spines['right'].set_color('none')
  ax.spines['top'].set_color('none')
  ax.xaxis.set_ticks_position('bottom')
  ax.yaxis.set_ticks_position('left')
  ax.spines['bottom'].set_position(('data',0))
  ax.spines['left'].set_position(('data',0))
  plt.ticklabel_format(style='sci', axis='x', scilimits=(0,0))
  plt.ylabel(r'Concentraion_$mol/m^3$')
  plt.xlabel(r'x_m')
  plt.legend(loc='best',edgecolor='none')
  plt.title("Anodic voltage:%.3fV; Equilibrium potential:%.3fV" % (E_a,E_0))
  plt.savefig('Concentration_profile_oxidation.png',dpi=300)
  plt.show()


CurrentVsTime(1.02,0.98,1,1)
ConcentrationProfileReduction(0.95,1,1)
```

```
ConcentrationProfileOxidation(1.05,1,1)
```

As a practice, you can simulate a model of anodic chronoamperametry (the solution only has species R initially). Also, you can change the Nernstian assumption to the more common assumption, Bulter-Volmer equation below.

$$j = j_0 \left\{ \frac{c_R(0,t)}{c_R^*} \exp\left[\frac{\alpha_a z F \eta}{RT}\right] - \frac{c_O(0,t)}{c_O^*} \exp\left[-\frac{\alpha_c z F \eta}{RT}\right] \right\} \tag{2.25}$$

In the i-E curve of Figure 2.3, we can see that the larger overpotential gives a higher initial current. Due to the limit of surface concentration (minimum concentration is zero), the limiting currents are same. With longer time on the applied potential, the diffusion thickness increases.

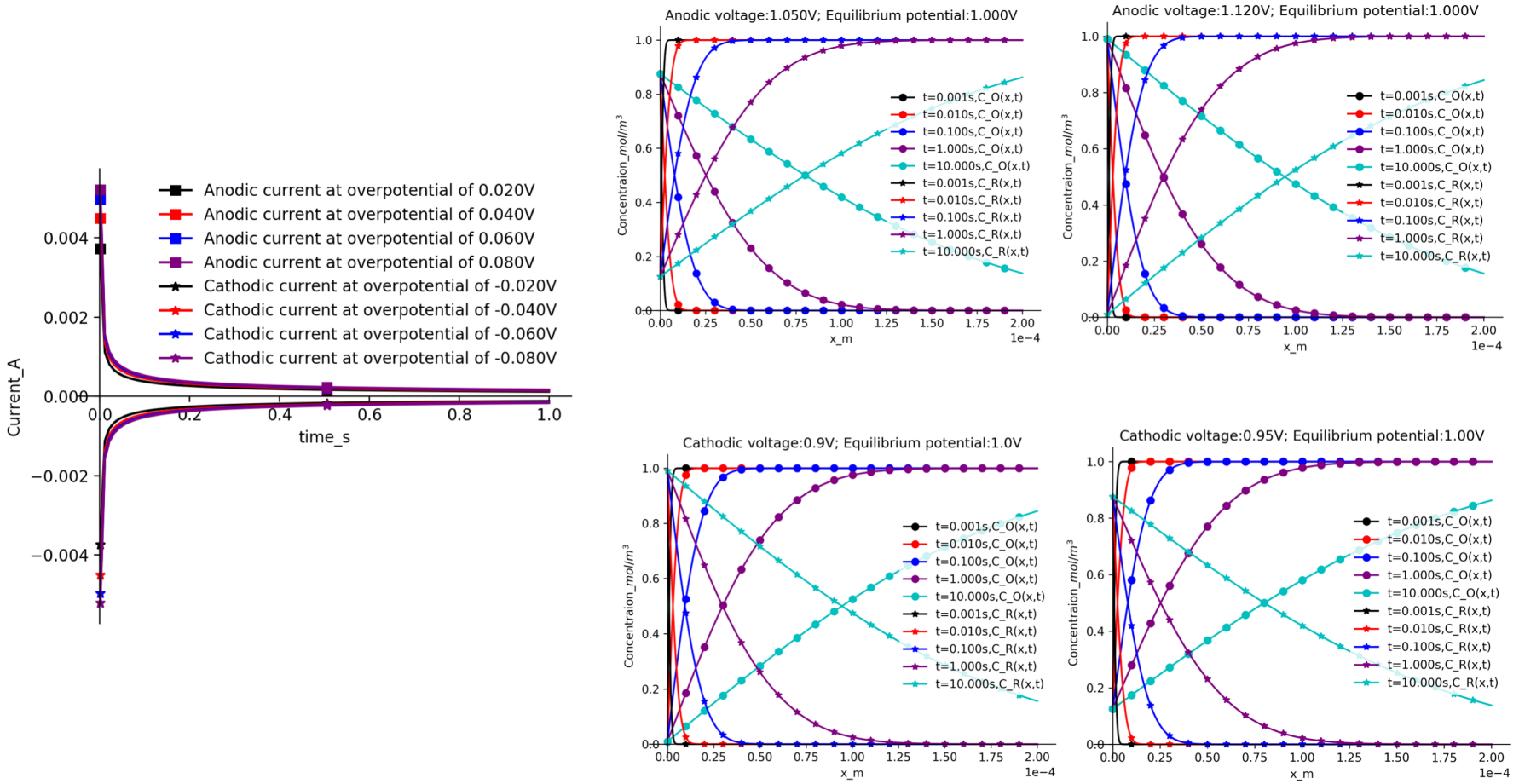

Figure 2.3: Current vs time of chronoamperametry and the concentration profile

{fig:chronoampera

2.2 Linear sweep voltammetry

In the chronoamperometry experiment, the potential was held as a constant value. The potential can also be swept linearly at v (V/s) and expressed as,

$$E(t) = E_i - vt \tag{2.26}$$

We still assume a completely reversible reaction. Then the boundary condition about the voltage and surface concentration changes to,

$$\begin{aligned} \frac{C_O(0,t)}{C_R(0,t)} &= exp[\frac{nF}{RT}(E_i - vt - E^{0'})] \\ &= \theta S(t), \text{ where } S(t) = exp(-\frac{nF}{RT}vt) \end{aligned} \tag{2.27}$$ {eq22}

The rest of boundary conditions are same with the numerical modeling of chronoamperametry. The Laplace transformed $C_O(x,t)$ and $C_R(x,t)$ are same with Equation 2.17 and 2.18. Since the flux balance is still same with Equation 2.19, we can get same relationship between $B(s)$ and $A(s)$ as Equation 2.20.

From Equation 2.19, one can get,

$$A(s) = \frac{\bar{i}(s)}{nFA\sqrt{sD_O}} \tag{2.28}$$ {eq23}

Combination of Equation 2.17, 2.18, 2.19, 2.20, and 2.28, one can get,

$$\begin{aligned} \overline{C_O}(x,s) &= \frac{C_O^*}{s} + \frac{\bar{i}(s)}{nFA\sqrt{D_O}\sqrt{s}} exp(-\sqrt{\frac{s}{D_O}}x) \\ \overline{C_R}(x,s) &= -\frac{\bar{i}(s)\epsilon}{nFA\sqrt{D_R}\sqrt{s}} exp(-\sqrt{\frac{s}{D_R}}x) \end{aligned} \tag{2.29}$$ {eq24}

When $x = 0$, Equation 2.29 can be rewrite as,

$$\begin{aligned} \overline{C_O}(0,s) &= \frac{C_O^*}{s} + \frac{\bar{i}(s)}{nFA\sqrt{D_O}\sqrt{s}} \\ \overline{C_R}(0,s) &= -\frac{\bar{i}(s)\epsilon}{nFA\sqrt{D_R}\sqrt{s}} \end{aligned} \tag{2.30}$$ {eq25}

In Equation 2.30, we can find there is $\bar{i}(s)$ and $1/\sqrt{s}$ which cannot be reversely transformed by the table in Figure 2.2. A convolution integral has to be used.

$$\begin{aligned} \mathscr{L}^{-1} f(s)g(s) &= F(t) * G(t) \\ &= \int_0^t F(t-\tau)G(\tau)d\tau \end{aligned} \tag{2.31}$$

Here * sign is NOT multiplication. It symbolizes the convolution integral. In Equation 2.30, $f(s) = \bar{i}(s)$ and $g(s) = 1/\sqrt{s}$. Inversion of $f(s)$ and $g(s)$ is $i(t)$ and $1/\sqrt{\pi t}$. Then we can carry the reverse Laplace transform on Equation 2.30,

$$\begin{aligned} C_O(0,t) &= C_O^* + \frac{1}{nFA\sqrt{D_O\pi}} \int_0^t i(\tau)(t-\tau)^{-\frac{1}{2}} d\tau \\ C_R(0,t) &= -\frac{1}{nFA\sqrt{D_R\pi}} \int_0^t i(\tau)(t-\tau)^{-\frac{1}{2}} d\tau \end{aligned} \tag{2.32}$$ {eq26}

Equation **??** is also not related with the assumption of reversible reaction. Now we can plug Equation 2.27 into Equation 2.32 to get,

$$\int_0^t i(\tau)(t-\tau)^{-\frac{1}{2}}d\tau = -\frac{nFAC_O^*\sqrt{\pi D_O}}{1+\theta S(t)\epsilon} \tag{2.33}$$ {eq27}

The general form of Equation 2.33 is $\int_0^t I(\tau)G(t-\tau) = F(t)$, where $I(\tau), \tau \in [0,t]$ is what we want to know and $F(t)$ is a known function. It has no analytical solution, then a numerical solution is needed. After $I(\tau), \tau \in [0,t]$ is known, we just need to plot $i(t)$ vs $E(t)$ then i-E curve can be obtained. Next part will involve some mathematical knowledge to solve the equation numerically.

When τ closes to t, we have trouble on the converge of $1/\sqrt{t-\tau}$. The left side of Equation 2.33 is a Riemann–Stieltjes integral which can be converted by an integration by parts,

$$\begin{aligned} F(t) = \int_0^t \frac{I(\tau)}{\sqrt{t-\tau}}\mathrm{d}\tau &= -2I(\tau)\sqrt{t-\tau}\Big|_0^t + 2\int_0^t I'(\tau)\sqrt{t-\tau}\mathrm{d}\tau \\ &= -2I(0)\sqrt{t} + 2\int_0^t I'(\tau)\sqrt{t-\tau}\mathrm{d}\tau \\ &= 2\int_0^t I'(\tau)\sqrt{t-\tau}\mathrm{d}\tau \end{aligned} \tag{2.34}$$ {eq28}

In the derivation above, integration by parts was used.

$$\int_a^b f(x)dg(x) = f(x)g(x)\Big|_a^b - \int_a^b f(x)g'(x)dx \tag{2.35}$$

Then we can numerically solve the right side of Equation 2.8. Here we can use the simplest method, midpoint rule ($\int_a^b = (b-a)f(\frac{a+b}{2})$), to express the integration numerically.

For $t \in [t_i, t_{i+1}]$,

$$\begin{aligned} \int_{t_i}^{t_{i+1}} I'(\tau)\sqrt{t-\tau}d\tau &\approx \Delta t I'(t_i + \frac{1}{2})\sqrt{\Delta t}\sqrt{n-i-\frac{1}{2}} \\ &\approx \Delta t\frac{I(t_{i+1})-I(t_i)}{\Delta t}\sqrt{\Delta t}\sqrt{n-i-\frac{1}{2}} \\ &= (I(t_{i+1})-I(t_i))\sqrt{\Delta t}\sqrt{n-i-\frac{1}{2}} \\ &= (I(t_{i+1})-I(t_i))\sqrt{\Delta t}p_i \end{aligned} \tag{2.36}$$ {eq29}

In practice, we can simulate the time from 0 to $100s$ with 1s as the interval (Δt). For a given t ($t = n\Delta t$), we can calculate F(t) from the explicit function of F(t) given by Equation 2.33. In the meanwhile, $t = n\Delta t$ on the right side of Equation 2.34. To numerically express the left side of Equation 2.33, we need to sum up Equation 2.36 from $i = 0$ to $n-1$.

$$\begin{aligned} F(n\Delta t) &= 2\sum_{i=0}^{n-1}\int_{t_i}^{t_{i+1}} I'(\tau)\sqrt{t-\tau}d\tau \\ &\approx 2\sum_{i=0}^{n-1}(I(t_{i+1})-I(t_i))\sqrt{\Delta t}p_i \\ &= 2\sqrt{\Delta t}[p_{n-1}I_n + \sum_{i=1}^{n-1}(p_{i-1}-p_i)I_i - p_0I_0)] \end{aligned} \tag{2.37}$$ {eq30}

Here, I_n is the current at $t = n\Delta t$.

$$I_n \approx \frac{1}{p_{n-1}}[\frac{F(n\Delta t)}{2\sqrt{\Delta t}} + p_0 I_0 - \sum_{i=1}^{n-1}(p_{i-1} - p_i)I_i] \tag{2.38}$$ {eq31}

We know the current equals to 0 with $t = 0$. Then we can use Equation 2.38 to calculate current with $t = \Delta t, n = 1. I(t) = (1/p_{n-1})(F(\Delta t)/2\sqrt{\Delta t} + p_0 I_0)$. Here $p_{n-1} = \sqrt{n-(n-1)-1/2} = \sqrt{1/2}$. We can alway use the current at the previous time to calculate the next t. Finally, we can get current vs time.

To increase the accuracy, we can use trapezoidal rule or partial midpoint method instead of midpoint method to numerically solve the integration. The Equation 2.37 still can be used just with different p_i.

Source code of Python was attached here. The i-E curve is also shown. After we know how to simulate linear sweep voltammetry. The cyclic voltammetry is easily to be modeled. We just need to add a backward scanning.

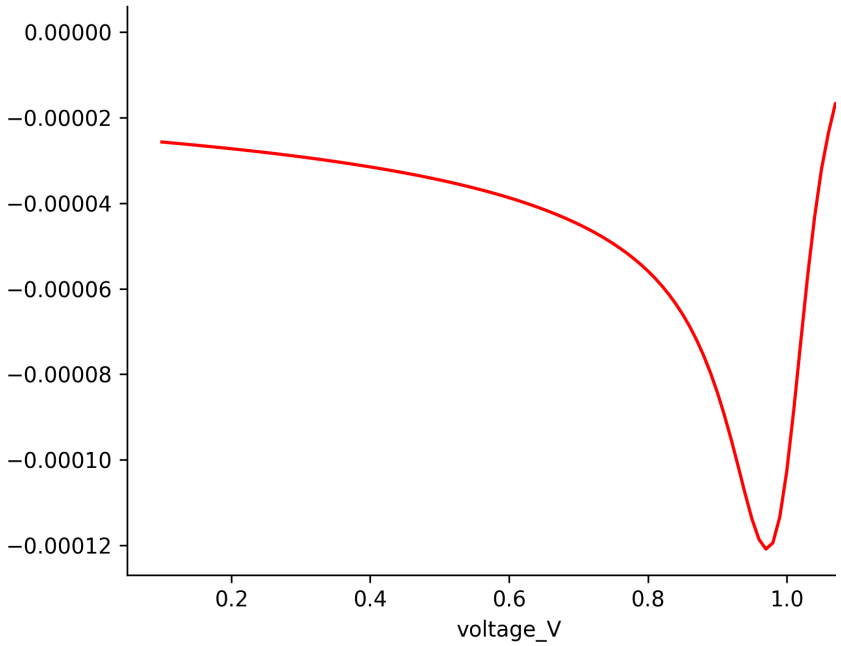

Figure 2.4: i-E curve of linear sweep voltammetry {fig:LSV}

```
# -*- coding: utf-8 -*-
from __future__ import division
import matplotlib.pyplot as plt
import numpy as np
from scipy import special
import math


C_OB = 1. #mol/m3
C_RB = 1. #mol/m3
D_O = 1E-9 #m2/s
```

```
D_R = 1E-9 #m2/s
F = 96485. #C/mol
R = 8.314 #J / mol. K
pi = 3.14
A = 1E-4 #1m2
T = 298. #K
epsilon = np.sqrt(D_O / D_R)
color=['black','red','blue','purple','c','grey','green']
S_t = []
F_t = []

def CurrentVsVoltage(E_i,E_0,n,v):
 #applied anodic voltage, equilibrium voltage, number of electronsscan rate (V/s)
  current = [] # current,I(t)
  voltage = [] # voltage,V(t)
  t = np.linspace(0, 50, 101)
  delta_t = (t[-1]-t[0])/(len(t)-1) #step of time
  theta = np.exp(n * F * (E_i - E_0) / R / T)
  epsilon = np.sqrt(D_O/D_R)
  S_t = np.exp(n*F*(-v*t)/R/T)
  F_t =  - n*F*A*C_OB * np.sqrt(pi*D_O) / (1. + theta*epsilon*S_t)
  s =  Pfuncion_Midpoint(1,0) * F_t[1]/(2.*np.sqrt(delta_t)) #current at t=t[1]
  voltage.append(E_i)
  voltage.append(E_i-v*delta_t)
  current.append(0) #initial current=0
  current.append(s)


  for i in range(2,len(t)):
    k = 1
    sum = 0
    while k <= i-1:
      sum = sum + (Pfuncion_Midpoint(i,k-1)-Pfuncion_Midpoint(i,k)) * current[k]
      k = k +1
    s = (1/Pfuncion_Midpoint(i,i-1)) * (F_t[i]/(2.*np.sqrt(delta_t))-sum)
    E = E_i - v*i*delta_t
    current.append(s)
    voltage.append(E)


  plt.figure()
  plt.plot(voltage,current,'-',color = 'red')
  ax = plt.gca()
  ax.spines['right'].set_color('none')
  ax.spines['top'].set_color('none')
  ax.xaxis.set_ticks_position('bottom')
  ax.spines['right'].set_color('none')
  ax.spines['top'].set_color('none')
  ax.xaxis.set_ticks_position('bottom')
```

```
  ax.yaxis.set_ticks_position('left')

  plt.ylabel(r'Current_A')
  plt.xlabel(r'voltage_V')
  plt.legend(loc='best',edgecolor='none')
  plt.savefig('current_vs_potential.png',dpi=300)
  plt.show()

def Pfuncion_PartialMidpoint(n,i):
  p = (2./3) * ((n-i)**1.5-(n-i-1)**1.5)
  return p
def Pfuncion_Midpoint(n,i):
  p = np.sqrt(n-i-1./2)
  return p
CurrentVsVoltage(1.1,1,1,0.02)
```

3 Hydrogen-Oxygen PEM fuel cell modeling

To be continued in Version 2.0. I plan to write it in the three subsections shown below.

3.1 LU decomposition of linear system

3.2 Physical model of PEM fuel cell

3.3 Numerical solution of PEM fuel cell