Solving a non-linear electrochemical current distribution problem using BAND in Python

Joshua Gallaway, Northeastern University Department of Chemical Engineering
10 Nov 2017

We wish to solve the boundary value problem of Tafel kinetics in a concentration-independent porous electrode in Cartesian coordinates. An analytical solution to this problem is given in "Theoretical Analysis of Current Distribution in Porous Electrodes," John S. Newman and Charles W. Tobias, *J. Electrochem. Soc.*, 109 (12) 1183-1191, 1962.[1] We will use Newman's BAND procedure to prepare a numerical solution, using Python.[2] This will allow us to compare the analytical and numerical solutions, to assure that our BAND computer code is correctly operational. This is an important step before moving on to more difficult problems that do not have an analytical solution.

1. Setting up the problem to be solved using BAND

The problem is defined in Newman and Tobias by equations [12], [14], [11], and [10]. The analytical solution is given in equation [17]. We will define the problem thus:

Conservation of charge:		
1	$\frac{di_1}{dx}+\frac{di_2}{dx}=0$	$x=L, i_1=I$
Tafel interfacial kinetics:		
2	$-\frac{di_2}{dx}=ai_0exp\left[-(1-\alpha)\frac{nF}{RT}(\phi_1-\phi_2)\right]$	$x=0, i_2=I$
Ohm's law for electronic current in the solid:		
3	$i_1=-\sigma\frac{d\phi_1}{dx}$	$x=0, i_1=\frac{d\phi_1}{dx}=0$
Ohm's law for ionic current in the liquid:		
4	$i_2=-\kappa\frac{d\phi_2}{dx}$	$x=0, \phi_2=0$

Here the sign convention is altered in Eq. 2, compared to Newman and Tobias. (Terms pertaining to this are highlighted.) The single spatial direction, x, is the distance across the electrode, with x = 0 at the separator and x = L at the current collector. The problem presents four equations and four unknown variables, all of which are functions of location:

$i_1=f(x)$	$i_2=f(x)$	$\phi_1=f(x)$	$\phi_2=f(x)$

To solve this problem we must first linearize Eq. 2, which has two of the unknown variables in an exponential term. We assume there is a nearly correct *trial value* of the variables, ϕ_1^0, and that this is separated from the current value by a small *change variable*, $\Delta\phi_1$. Thus:

$\phi_1=\phi_1^0+\Delta\phi_1$	$\phi_2=\phi_2^0+\Delta\phi_2$

We substitute these into Eq. 2, ignoring any terms quadratic in the change variables. Then we substitute back:

$\Delta\phi_1 = \phi_1 - \phi_1^0$	$\Delta\phi_2 = \phi_2 - \phi_2^0$

The linearized version of Eq. 2 is thus:

$$-\frac{di_2}{dx} - ai_0 exp[\beta(\phi_1^0 - \phi_2^0)]\beta\phi_1 + ai_0 exp[\beta(\phi_1^0 - \phi_2^0)]\beta\phi_2 = ai_0 exp[\beta(\phi_1^0 - \phi_2^0)][1 - \beta\phi_1^0 + \beta\phi_2^0]$$

Where:

$$\beta = -(1-\alpha)\frac{nF}{RT}$$

To simplify our job writing equations we further define:

$$P = ai_0 exp[\beta(\phi_1^0 - \phi_2^0)]$$

Finally giving us:

Tafel interfacial kinetics:		
2	$-\frac{di_2}{dx} - P\beta\phi_1 + P\beta\phi_2 = P[1 - \beta\phi_1^0 + \beta\phi_2^0]$	$x = 0, i_2 = I$

Now we are prepared to cast the equations into the general linear form defined by Newman:

$$\sum_k a_k \frac{d^2c_k}{dx^2} + b_k\frac{dc_k}{dx} + d_k c_k = g$$

Here the subscript k is the number of unknown variables in the problem, and all the variables are denoted by “c.” The constants a, b, d, and g are the coefficients of the variables, their first and second derivatives, and the inhomogeneous coefficient. This is just a general way of writing an equation that can contain all possible variable terms.

To use Newman’s terminology we rename:

$c_1 = i_1 = f(x)$	$c_2 = i_2 = f(x)$	$c_3 = \phi_1 = f(x)$	$c_4 = \phi_2 = f(x)$

And now the problem is recast:

Conservation of charge:		
1	$c_1' + c_2' = 0$	$x = L, c_1 = I$
Tafel interfacial kinetics:		
2	$-c_2' - P\beta c_3 + P\beta c_4 = P[1 - \beta\phi_1^0 + \beta\phi_2^0]$	$x = 0, c_2 = I$
Ohm's law for electronic current in the solid:		
3	$c_1 + \sigma c_3' = 0$	$x = 0, c_1 = c_3' = 0$
Ohm's law for ionic current in the liquid:		
4	$c_2 + \kappa c_4' = 0$	$x = 0, c_4 = 0$

Here we can write the coefficients for each equation:

Eq. 1:

$$\begin{array}{c|cccc} & a & b & d & g \\ \hline c_1 & 0 & 1 & 0 & 0 \\ c_2 & 0 & 1 & 0 & \\ c_3 & 0 & 0 & 0 & \\ c_4 & 0 & 0 & 0 & \end{array}$$

Eq. 2:

$$\begin{array}{c|cccc} & a & b & d & g \\ \hline c_1 & 0 & 0 & 0 & P[1 - \beta\phi_1^0 + \beta\phi_2^0] \\ c_2 & 0 & -1 & 0 & \\ c_3 & 0 & 0 & -P\beta & \\ c_4 & 0 & 0 & P\beta & \end{array}$$

Eq. 3:

$$\begin{array}{c|cccc} & a & b & d & g \\ \hline c_1 & 0 & 0 & 1 & 0 \\ c_2 & 0 & 0 & 0 & \\ c_3 & 0 & \sigma & 0 & \\ c_4 & 0 & 0 & 0 & \end{array}$$

Eq. 4:

$$\begin{array}{c|cccc} & a & b & d & g \\ \hline c_1 & 0 & 0 & 0 & 0 \\ c_2 & 0 & 0 & 1 & \\ c_3 & 0 & 0 & 0 & \\ c_4 & 0 & \kappa & 0 & \end{array}$$

Now we will solve the problem using the BAND function in Python. We have four equations and four unknowns, and will break up the distance x into 100 discrete segments, so we define:

```
N = 4
NJ = 100
```

The tables above will be duplicated at all 100 points in the domain. The way BAND expects this information is N x NJ x N matrices for a, b, and d, and an N x NJ matrix for g. In Python, we initialize the matrices:

```
sma = zeros([N,NJ,N])
smb = zeros([N,NJ,N])
smd = zeros([N,NJ,N])
smg = zeros([N,NJ])
```

The first index represents the equation number. The second index is the position. The third index is the unknown variable number. Now we must transcribe the non-zero coefficients above into the appropriate spots in the matrices. As an example we consider smb, which contains the values of b for all positions. Thus:

```
smb[0,:,0] = 1
smb[0,:,1] = 1
smb[1,:,1] = -1
smb[2,:,2] = sigma
smb[3,:,3] = kappa
```

Recall that Python indexes from zero instead of one, so the top command sets the value of b for the first variable in the first equation to 1 at all NJ positions. The second command does the same for the second variable in the first equation. Et cetera. This is done until all the non-zero a, b, d, and g values are entered. To visualize this, we have defined a matrix:

$$smb = \begin{array}{cccc|l} 1 & 1 & 0 & 0 & \text{Eq. 1} \\ 0 & -1 & 0 & 0 & \text{Eq. 2} \\ 0 & 0 & \sigma & 0 & \text{Eq. 3} \\ 0 & 0 & 0 & \kappa & \text{Eq. 4} \\ c_1 & c_2 & c_3 & c_4 & \end{array}$$

This matrix has NJ = 100 identical slices into/out of the page, and it tells us the coefficients of the first derivatives of all the unknown variables at all positions. However, the equations will be different at the boundaries, which are position indexes 0 and NJ-1. The general form for the boundary conditions is:

$$\boxed{\sum_k p_k \frac{dc_k}{dx} + e_k c_k = f}$$

We use this expression to build matrices smp, sme, and smf, which then replace smb, smd, and smg at the boundaries. Three of the boundary conditions are at x = 0 (position index 0), which we will call B.C. 1:

B.C. 1 (x = 0):

$$smp = \begin{array}{cccc} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ c_1 & c_2 & c_3 & c_4 \end{array} \qquad sme = \begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ c_1 & c_2 & c_3 & c_4 \end{array} \qquad smf = \begin{array}{c|l} 0 & \text{Eq. 1} \\ I & \text{Eq. 2} \\ 0 & \text{Eq. 3} \\ 0 & \text{Eq. 4} \end{array}$$

Here the gray values are just the regular entry for Eq. 1, and the non-gray are the boundary conditions. One of the boundary conditions is at x = L (position index NJ-1), which we will call B.C. 2:

B.C. 2 (x = L):

$$\text{smp} = \begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma & 0 \\ 0 & 0 & 0 & \kappa \\ c_1 & c_2 & c_3 & c_4 \end{array} \qquad \text{sme} = \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ c_1 & c_2 & c_3 & c_4 \end{array} \qquad \text{smf} = \begin{array}{cc} I & \text{Eq. 1} \\ 0 & \text{Eq. 2} \\ 0 & \text{Eq. 3} \\ 0 & \text{Eq. 4} \end{array}$$

2. Solving the nonlinear problem using BAND

The procedure takes the sma, smb, smd, and smg matrices defined above, manipulates them into two matrices that define the boundary value problem (ABD and G), and solves them using BAND. For a linear problem, a solution in one pass is assured.

Since our problem is nonlinear, it cannot be solved outright in a single step. Instead, we will iterate to a solution. The variables in the nonlinear terms appear in the matrices in two ways:

unknowns: c_3 and c_4	trial values: ϕ_1^0 and ϕ_2^0

Iteration will be successful when these values converge within a given tolerance. We begin by entering reasonable guesses for the trial values, the problem is solved using BAND, and the outputs become the trial values for the next iteration.

- Define the number of unknowns/equations by setting `N`.
- Define the number of spatial points the problem is divided into by setting `NJ`.
- Set the convergence tolerance `tol` and the maximum number of iterations `itmax`.
- Fill in the various constants for the problem (I, L, a, etc.).
- Within the `FILLMAT` function, enter the values of the sma, smb, smd, and smg matrices. Also enter smp, sme, and smf for both B.C. 1 and B.C. 2. The code will automatically insert these at the domain boundaries.
- Within the `INITGUESS` function put in reasonable initial guesses. These are trial values of the four unknown variables c_1 through c_4 at all spatial indices 0 through NJ-1. Thus they are stored in an N x NJ matrix `cold` (which stands for "c-old" or the previous iteration's values of c).
- Run the code.

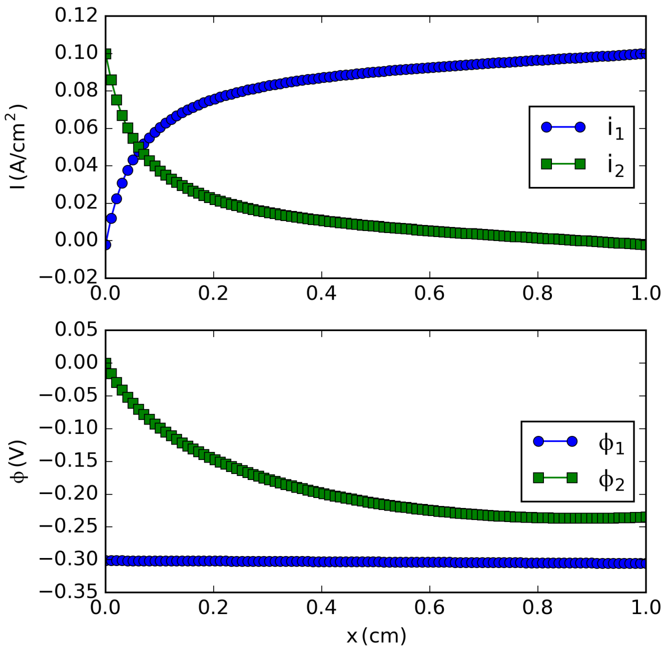
The steps the code will take are:

1. Create `sma`, `smb`, `smd`, and `smg` using `cold` by calling `FILLMAT`.
2. Create `ABD` and `G` using `sma`, `smb`, `smd`, and `smg` by calling `ABDGXY`.
3. Solving for the values of c (`delc`) using `ABD` and `G` by calling `BAND`.
4. Calculate an error by subtracting `delc` from `cold`.
5. Set `cold = delc` for the next iteration.
6. If the maximum error is below `tol` the program will consider the problem converged. If not, it will return to step 1.

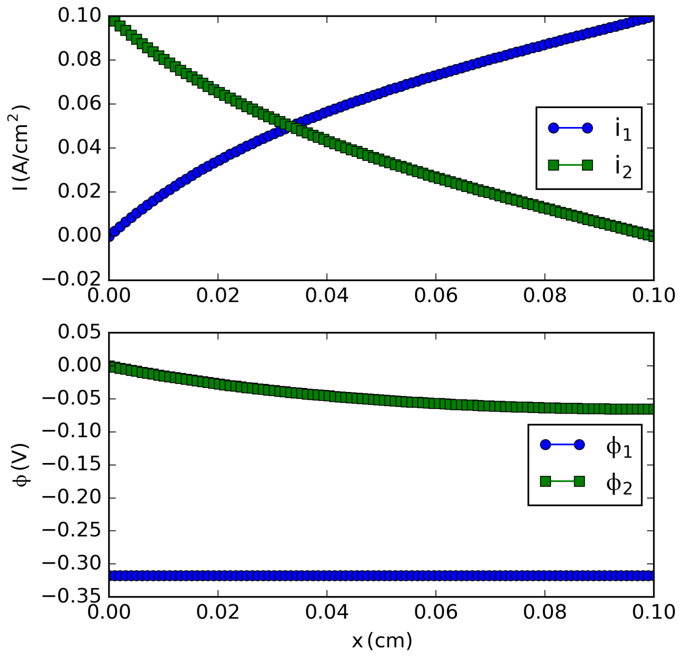
As a test case, let us consider these constants:

I	0.1 A/cm^2
a	23,300 cm^{-1}
i_0	2 × 10^{-7} A/cm^2
n	1 eq/mol
F	96,500 C/eq
R	8.314 J/mol-K
T	298 K
σ	20 S/cm
κ	0.06 S/cm
α	0.5

Note that it is the disparity between σ and κ that results in a distribution of current across the electrode thickness. If we select an electrode thickness of L = 1 cm, the result is as follows:



If we select an electrode thickness of L = 1 mm, the result is as follows:



References

1. Newman, J. S. and C. W. Tobias, "Theoretical Analysis of Current Distribution in Porous Electrodes," *J. Electrochem. Soc.*, 109 (12) 1183-1191, 1962.

2. Newman and Thomas-Alyea, Appendix C "Numerical Solution of Coupled, Ordinary Differential Equations" in: *Electrochemical Systems*, Third Edition, John Wiley & Sons, Hoboken, NJ, 2004.