Background of this system of PDEs and how boundary conditions matrixs are produced

Article:

Gupta N, Gattrell M, MacDougall B. Calculation for the cathode surface concentrations in the electrochemical reduction of CO2 in KHCO3 solutions[J]. Journal of applied electrochemistry, 2006, 36(2): 161-172.

<https://link.springer.com/article/10.1007/s10800-005-9058-y>

The initial PDEs are as follows, which describes the diffusion and reaction kinetics for species involved in CO2 electrochemistry reduction:

\*Red ‘1’ was missed by article author by accident.

Here, u1 to u4 represents concentration of CO2 (aq), HCO3-, CO32- and OH-, respectively.

Initial conditions of this PDEs are:

Boundary conditions are: (x = 0 is the electrode surface, x = 0.01 is the right boundary (chemically, x = 0.01 is large enough so that c(x) = c(bulk)). This is inversed against that defined in article, so that fliplr(x) is used in plotting for better comparation)

Constants are:

Du1 = ;

Du2 =;

Du3 =;

Du4 =;

k1f =;

k1r =;

k2f =;

k2r =;

All constants are transferred to proper units so that you can further read this document even you don’t care about chemistry background. For those who focus on its chemistry background, I will suggest you read the original paper.

The code here helps solve this system of PDEs, which largely relies on

https://www.mathworks.com/help/matlab/math/solve-system-of-pdes.html

, which is very detailed for you to understand other parts of the code.

But here, I would like to focus on how I produce Boundary conditions, which is not quite clearly explained and bothers me a lot when I’m working on it.

The standard form for Boundary Condition is:



For

We can transform it to:

Which is equivalent to

We can find that p(x,t,u) = , and as , q(x,t) can actually be any value. I set it as 0 in my code.

Similarly, for

We can transform it to:

This implies that p(x,t,u) = , q(x,t) = 1 by compare.

With these procedures, we can easily construct the matrix in our code:

pl = [(-1.45e-7);0;0;(1.45e-7)];

ql = [1;1;1;1];

pr = [ur(1)-0.0342;ur(2)-0.499;ur(3)-(7.6e-4);ur(4)-(3.3e-7)];

qr = [0;0;0;0];