

Introduction to dualfoil.f

Karen E. Thomas, Marc Doyle, and John Newman

October 2002

Contents

1	Introduction	1
2	Input file	2
2.1	Example for the use of lcurs, cu, tt, and mc:	5
2.2	Balancing electrode capacity	6
3	Output file	6
4	Common problems	7
5	Adding your own materials to the library	7
6	Modifying dualfoil	8

1 Introduction

We hope that this document will help answer questions you may have about how to use dualfoil.f. It tries to answer questions which we have received frequently in the past. Please keep in mind that we distribute this program for free.

dualfoil.f simulates the behavior of lithium and lithium-ion cells. The program is based on a continuum-scale model using the principles of concentrated solution theory, porous electrode theory, Ohm's law, Butler-Volmer kinetics, and current and mass conservation. It solves six coupled, nonlinear equations to calculate the cell voltage and current as a function of time, and to calculate the potentials in the electrolyte and electrode phases, salt concentration, intercalated lithium concentration, reaction rate, and current density in the electrolyte as functions of time and position within the cell. For more information on the governing equations, see

C. Marc Doyle, "Design and Simulation of Lithium Rechargeable Batteries", PhD Dissertation, University of California, Berkeley (1995).

M. Doyle, T. F. Fuller, and J. Newman, J. Electrochem. Soc., 140, 1526 (1993).

T. F. Fuller, M. Doyle, and J. Newman, J. Electrochem. Soc., 141, 1 (1994).

K. E. Thomas, J. Newman, and R. M. Darling, "Modeling of Lithium Batteries," in: *Advances in Lithium-Ion Batteries*, B. Scrosati and W. van Schalkwijk, eds., New York : Kluwer Academic Publishers (2002).

The model is one-dimensional, and thus assumes that the cell is uniform in the directions parallel to the current collectors. All quantities are therefore normalized to a per unit area basis. The load of a particular application will determine the area needed, and therefore that area is not included in the simulation. Note that for a given load, one could use a smaller cell with a higher applied current density or a larger cell with a lower applied current density. Changing the current density in the input file thus allows one to evaluate the optimum size of cell to power a given load.

Radial diffusion within particles of insertion material is treated using Duhamel's superposition integral, which assumes that the diffusion coefficient is constant. All other properties can be specified to vary with concentration and temperature. The cell temperature can vary with time, but temperature gradients within the cell are considered negligible. Presently, dualfoil's library of material properties does not include the temperature dependence of material properties, and these must be added by the user.

2 Input file

A sample input file is provided on the website. This input file very roughly approximates a typical commercial lithium-ion battery, but should not be construed to give the exact representation of any particular battery. In particular, the geometry of a battery is usually optimized for its design discharge time, and thus the electrode thicknesses, porosities, and particle radii vary considerably based on a battery's application. It is the responsibility of the user to input the geometry appropriate for the user's application and to ensure that the capacities of the positive and negative electrodes are balanced (see below).

A brief explanation of the input parameters follows, in the order in which they are listed in the input file.

- | | |
|----------|--|
| lim | number of iterations before the program decides that it cannot converge. 50 is usually sufficient |
| h1,h2,h3 | thicknesses of the negative electrode, separator, and positive electrode, respectively, in meters |
| hcn,hcp | thicknesses of the negative and positive current collectors. These are used only for calculating the total cell mass. |
| n1,n2,n3 | number of nodes (discretization or mesh points) used for numerical solution of the governing differential equations. Using more mesh points increases simulation accuracy but also increases computation time. The sum of $n1 + n2 + n3$ should not exceed |

220. If n1 is 0, then the program treats the negative electrode as a nonporous foil electrode, ignores x, dfs1, Rad1, ep1, epp1, and epf1 (see below), and uses h1 only to calculate the total cell mass.

T	initial temperature in kelvin
xi(1,1)	initial salt concentration in the electrolyte in mol/m ³
x,y	initial stoichiometry of lithium in the negative and positive electrodes, respectively. E.g., y in Li _y Mn ₂ O ₄ . The formula unit should be chosen so that y does not exceed 1. These parameters set the initial states of charge of the electrodes. Lithium stoichiometry increases in the positive electrode and decreases in the negative electrode during discharge.
tmmax	maximum time step size in seconds. The program starts with a time step size of 1 second, and if it converges sufficiently quickly, then it doubles this time step until the time step equals tmmax. Use a lower value of tmmax for high currents or difficult simulations, and a higher value to reduce computation time.
vcut	Cell voltage at which program simulation will halt, in volts.
dfs1,dfs3	Diffusion coefficients in the negative and positive insertion materials, respectively, in m ² /s. dualfoil.f does not allow for the diffusion coefficient to vary with state of charge.
Rad1,Rad3	average radius of the negative and positive active material, respectively, in meters.
ep1,ep2,ep3	volume fraction of electrolyte (i.e., porosity) in the negative electrode, separator, and positive electrode, respectively. ep2 may be less than 1 if an inert separator material is used, such as Celgard.
epp1,epp2,epp3	volume fraction of polymer. This parameter is meaningful only for use with gel electrolytes.
epf1,epf3	volume fraction of inert filler, such as conductive additives and binder.
sig1,sig3	electronic conductivity of the negative and positive composite electrodes, respectively, in S/m (S=ohm ⁻¹). This value should be that of a nonporous composite of the active material, binder, and conductive filler. The code corrects this value for tortuosity and porosity by multiplying it by (1-ep-epp-epf) ^{1.5} .
rka1,rka3	kinetic rate constants for reaction at the negative and positive electrodes, respectively, in mol/m ² s (i.e., exchange current density divided by Faraday's constant), evaluated at 1 molar electrolyte concentration and x or y = 0.5. dualfoil.f assumes that $\alpha_a=\alpha_c=0.5$.

$r_{anode}, r_{cathode}$ film resistance due to the solid-electrolyte interphase in ohm m^2 . This resistance is treated as acting in series with the charge-transfer resistance.

$cot1, cot3$ coulombic capacity of the negative and positive insertion materials, respectively, in mAh/g . These are calculated from the formula weight, assuming x or y ranges from 0 to 1.

r_e density of the electrolyte in kg/m^3 . Used only to calculate the total cell mass.

$rs1, rs3$ density of the negative and positive insertion materials, respectively, in kg/m^3 . These are used to calculate the total cell capacity based on the input volume fractions ($1 - \epsilon_p - \epsilon_{pp} - \epsilon_{pf}$).

$r_f, r_{pl}, r_c, r_{cn}, r_{cp}$ densities of the inert filler, polymer in a gel electrolyte, inert separator, and current collectors, respectively, in kg/m^3 . These values are only used to calculate the total cell mass.

htc effective heat-transfer coefficient around the cell ($\text{W/m}^2\text{K}$)

$dUdT$ entropy of reaction for the cell, in volts/Kelvin. This can also be specified as a function of stage of charge for each electrode by modifying subroutine `ekin` (see below)

C_p overall heat capacity of the cell in J/kg K

T_{am} ambient temperature in kelvin, assumed to be constant

n_{cell} number of cells in a stack, used for treating heat transfer in a multicell stack.

lht flag to set the mode of thermal operation. Use 0 for variable temperature, 1 for isothermal operation with the rate of heat generation calculated, and 2 for isothermal operation without calculation of the rate of heat generation. For adiabatic operation, set lht to 0 and htc to 0.0d0.

$il1, il2, il3$ flags to determine the output file. Setting $il1$ to 0 produces a short output file which contains only the cell voltage, current, utilization, and heat generation at each time step. Setting $il1$ to 1 prints additionally the concentration and potential profiles at each $il3$ th time step and $il2$ th node (see below).

$lflag$ generally use 1. Use 0 if initially there is salt only in the separator.

$lpow$ Flag to activate special peak power feature. Turned off if $lpow$ is 0. If set to 1, then at the end of the simulation the code will calculate the maximum amount of power the cell is capable of generating for a 30 second current pulse, based on the condition of the cell at the end of the simulation.

- jsol Flag to activate special solid concentration profile feature. Turned off if jsol is 0. Otherwise, at the end of the simulation the code will calculate the relative radial concentration profile of inserted lithium in an average particle located at the node specified by the value of jsol.
- nneg,nprop,npos flag to choose the chemistry of the negative active material, electrolyte, and positive active material, respectively, from the library in dualfoil. A guide to the library is given at the end of the sample input file provided on the website. The library contains materials properties which we have measured ourselves or gleaned from the literature, for materials which we have had interest to study. It is *not* a comprehensive library for all materials used in commercial batteries. See below on how to add your own materials.
- lcurs number of steps to simulate in the experiment (see below)
- cu(i) if mc(i) = 1 or 2, then cu(i) is the current for that galvanostatic step in A/m². A positive current will discharge the cell. if mc(i) = 0, then cu(i) is the cell potential in volts for that potentiostatic step.
- tt(i) if mc(i) = 0 or 1, then tt(i) is the time for the duration of the step in minutes. if mc(i) = 2, then tt(i) is the cutoff potential in volts
- mc(i) mode of operation for the ith step. see below.

2.1 Example for the use of lcurs, cu, tt, and mc:

Let's say you want to charge the cell at 10 A/m² until it hits a cutoff voltage of 4.3 V, then hold it at 4.3 V for 20 minutes, and then turn off the current and watch how the cell potential relaxes for 30 minutes. Then the last lines of the input file would be:

```
3      ! lcurs
-10.0d0 4.3d0 2
4.3d0 20.0d0 0
0.0d0 30.0d0 1
```

2.2 Balancing electrode capacity

The theoretical capacity of an electrode in Ah/m² is given by

$$h*(1-ep-epp-epf)*cot*rs$$

where h is the electrode thickness, $(1-ep-epp-epf)$ is the volume fraction of active material in the electrode, cot is the theoretical capacity of the active material based on x or y ranging from 0 to 1, and rs is the density of the active material. To get the actual capacity, multiply the theoretical capacity by the actual range of x or y (with the formula unit defined so that this range is always less than 1). Ranges for various materials are listed in the description of the library in the input file.

In a balanced cell, the actual capacities of each electrode are equal. Adjust the values of $h1$ and $h3$ to achieve a balanced cell, or to make one electrode limit the achievable capacity of the cell. Note that the code does not calculate any irreversible consumption of lithium due to side reactions.

3 Output file

As mentioned in the description of input parameter $ill1$, the code can produce a short output or a short output with long output. The short output depends on the choice of lht . For $lht = 0$, the output columns (left to right) are utilization, cell potential in volts, temperature in degrees Celsius, time in minutes, open-circuit potential in volts, current in A/m², total rate of heat generation in W/m², and rate of irreversible heat generation in W/m². For $lht = 1$, the output is utilization, cell potential, heat transfer coefficient that would be required to keep the cell isothermal in W/m²K, time, open-circuit potential, current, total rate of heat generation, and rate of irreversible heat generation. For $lht = 2$, the output is utilization, cell potential, time, open-circuit potential, current, and the net specific energy up to that time.

The open-circuit potential reported in the short output is the potential to which the cell would eventually relax if the current were shut off at that point in time. The difference between the open-circuit potential and the cell potential thus is the total overpotential of the cell at that point in time.

The total rate of heat generation includes both resistive (ohmic, kinetic, and concentration overpotential) and entropic terms (if data for entropy are included in the input file under dU/dT or in the materials library as a function of state of charge). The irreversible heat generation includes only resistive heat generation.

Utilization is the average value of the lithium stoichiometry in the positive electrode. For a balanced cell, $(utilization - y_{min})/(y_{max}-y_{min})$ is the state of charge of the cell, with 1 being fully discharged, where y_{min} is the minimum and y_{max} is the maximum lithium concentration attainable in the particular insertion material. Values of y_{min} and y_{max} for various electrode materials

are given in the input file in the description of the library. If the electrodes are not balanced and the capacity of the negative electrode is lower than the capacity of the positive electrode, then the “state of charge” of the cell would be more accurately determined by the utilization of the negative electrode.

If `il1` is set to 1, then the output file will also include the long output, which gives profiles of the dependent variables across the cell. `(timesteps)/il3` profiles will be printed, where `timesteps` is the total number of time steps simulated in the run. The output consists of seven columns at each time step: distance, salt concentration, potential in the electrolyte, `x` or `y`, current density in the electrolyte, reaction rate, and potential in the electrode. Distance across the cell is in micrometers, with zero being the interface between the negative current collector and the negative electrode. `x` is the lithium stoichiometry in the negative electrode while `y` is the lithium stoichiometry in the positive electrode, both evaluated at the surface concentrations of particles.

4 Common problems

There are two common reasons why the program might not converge. If the program does not converge, it will automatically print out the long output, which can be used to determine the reason for nonconvergence. The first common reason is that the user has specified a current higher than the electrolyte is capable of bearing, and the salt concentration has been driven to zero in the cathode. In the long output, look at the salt concentration profile at the last time step to see if it has been driven to zero. Remedy the problem by using a lower current density or higher electrolyte porosity. The second common reason is that the user is trying to overcharge or overdischarge a cell. This creates problems for two reasons. The first is that the open-circuit potential often changes very sharply with state of charge when the cell is overcharged or overdischarged, and the change may be so steep that the program cannot converge. The second is that `x` or `y` cannot be lower than 0 or greater than 1 in the program (because of how the electrode kinetics are treated), and attempts to force the lithium stoichiometry outside of these ranges may produce a “divide by zero” error. In the long output, look at the `x` or `y` column to determine if the cell had reached full charge or full discharge. Remedy the problem by changing the voltage cutoff, reducing the time of charge or discharge, reducing the current, or increasing the electrode thicknesses.

5 Adding your own materials to the library

The solid diffusion coefficient, exchange current density, electronic conductivity of the composite electrode, film resistance, coulombic capacity of the active material, and density of the active material must be entered by the user in the input file (see input file section above). `dualfoil.f` contains a library of open-circuit potential as a function of state of charge for a variety of positive and negative electrode materials and also a library of electrolyte properties. You can choose materials from

the library by the value of npos, neg, and nprop in the input file. In addition, there is space in the library for you to add your own materials.

New electrode materials are added to subroutine ekin, in the space marked “add your own”. Add a functional form for the open-circuit potential (called g0) as a function of intercalated lithium concentration in mol/m³ (called xx(3,j)) or lithium stoichiometry (xx(3,j)/ct1 for negative electrodes, xx(3,j)/ct3 for positive electrodes). Then add g1, which is the derivative of g0 with respect to xx(3,j).

New electrolytes are added in subroutine prop, in the space marked “add your own”. Properties can be a function of salt concentration, given by xx(1,j) in mol/m³. df is the salt diffusion coefficient in m²/s, cd is the ionic conductivity in S/m, tm is the transference number of the cation (t_+), and dfu is $\frac{d \ln f_{\pm}}{dc}$, where f_{\pm} is the mean molar activity coefficient of the salt and c is salt concentration in mol/m³. ddf, dcd, dtm, and d2fu are the derivatives of df, cd, tm, and dfu, respectively, with respect to xx(1,j). You can also add a dependence on temperature T. The diffusion coefficient and conductivity are corrected for tortuosity and porosity by multiplying by $\epsilon^{1.5}$.

6 Modifying dualfoil

We strongly advise against modifying the code unless you are knowledgeable about battery simulation. In particular, modifying subroutines comp, band, matinv, and calca could seriously mess up the program. In this section we describe where it is safe to modify subroutines.

The format of the short output file is specified in subroutine cellpot, in format lines 309 or 307 depending on the value of lht (see output file section).

The format of the long output file is specified in subroutine nucamb. Dependent variables are stored at each time step in xt(i,j,l), where l is the time step, j is the node, and i specifies the variable according to the following: 1 is salt concentration in the electrolyte, 2 is potential in the electrolyte phase, 3 is concentration of inserted lithium, 4 is current density in the electrolyte, 5 is reaction rate, and 6 is potential in the electrode phase.

Time-stepping is controlled in main. Comments in the code (search for “time step”) indicate which lines to comment out in order to have the code run with constant time steps. For extremely stiff problems, try using a smaller initial time step.

Currently, one cutoff potential is entered in the input file, and the code will automatically stop when it hits that cutoff potential (whether on charge or discharge). This creates problems when trying to simulate the cycling of a cell. To turn this off and control the cutoffs solely from mc, tt, and cu, comment out the line marked with “vcut option” in main.

To use more than 220 nodes, increase the array sizes of the variables in common blocks ssblock, var, and tprop, and in the dimension statements for term and termn.

If lcurs is greater than 15, then the code assumes that you are trying to simulate cyclic voltammetry, and will cycle indefinitely until you terminate the program. If you want to change this, comment out the if statement beginning with the line “If(L.EQ.LCURS .AND. LCURS.GE.15)”.

To allow lcurs to be greater than 200, change the array sizes of mc, cu, and tt..