# Microstructural Modeling of Lithium Battery Electrodes

by

Benjamin Hellweg

B.S. Materials Science and Engineering
Massachusetts Institute of Technology, 1997

Submitted to the Department of Materials Science and Engineering
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Materials Science and Engineering

at the

Massachusetts Institute of Technology

September 2000

Signature of Author..............................................
Department of Materials Science and Engineering
August 4, 2000

Certified by.............................................. YM Chiang
Yet-Ming Chiang
Kyocera Professor of Ceramics
Thesis Supervisor

Accepted by.............................................. Carl V.
Carl V. Thompson
Stavros S. Salapatas Professor of Materials Science and Engineering
Chairman, Departmental Committee on Graduate Students

# MICROSTRUCTURAL MODELING OF LITHIUM BATTERY ELECTRODES

by

BENJAMIN HELLWEG

Submitted to the Department of Materials Science and Engineering
on August 4, 2000 in partial fulfillment of the
requirements for the Degree of Master of Science in
Materials Science and Engineering

## ABSTRACT

The transport of charged species in lithium ion batteries was studied from a microstructural point of view. Electron transport was analyzed using percolation theory and comparison with other conductor-insulator composites. An *in situ* filter pressing apparatus was designed and constructed in order to determine the percolation threshold in composite electrode systems. In addition, the effect of inter-particle interactions was qualitatively examined. The percolation threshold was determined to occur between 10 and 13 volume percent conductor loading for liquid electrolyte systems. In dissolved polymer systems, polymer adsorption shifted the percolation threshold to 25 volume percent.

Ion transport was analyzed using a computer model designed by Doyle and Newman. Microstructural solutions to ameliorate the rate limiting steps were proposed and tested. Battery simulations demonstrated that the rate capability of lithium batteries could be improved both by utilizing plate-like particles aligned in parallel with the current flow, and also by producing a porosity gradient in the electrode. Using particles aligned parallel to the current flow allowed the elimination of tortuosity from the ion path. Graded electrodes provided superior ion transport near the electrode surface, where the ionic current is greatest, while additional capacity was available in the depth of the eletrode, where ion transport was not as critical.

Thesis Supervisor: Yet-Ming Chiang

Title: Kyocera Professor of Ceramics

## Table of Contents

## List of Illustrations and Figures

## List of Tables

## Acknowledgements

First and foremost, I would like to express my sincere thanks to Prof. Yet-Ming Chiang for serving as my thesis advisor. It is in no small part due to his patience, encouragement, advice and expertise that this thesis came about.

Many thanks also to my parents for their unconditional love, support and encouragement.

Thanks my colleagues in the Chiang research group, especially those working on batteries, Pimpa Limthongkul, Haifeng Wang, Garry Maskaly and Young-Il Jang. Their help and questions were invaluable.

Thanks to Prof. Craig Carter and Catherine Bishop for their help with modeling and numerical methods.

Thanks to Phil Soo for providing precious BCE samples for use in filter-pressing experiments.

Thanks to all of my colleagues and faculty in the Department of Materials Science and Engineering.

Thanks to the Department of Defense for Fellowship support.

The resources of the CMSE shared experimental facilities are gratefully acknowledged.

# 1. Introduction

## 1.1 Introduction

Lithium ion batteries are the system of choice for meeting the increasing power demands of today's portable consumer electronics. With continuously increasing need for portable power, a large research effort has been directed at decreasing the cost of manufacturing batteries while increasing their energy and power densities.

The typical lithium battery consists of a lithium foil or a composite carbon anode, a liquid electrolyte containing lithium salt, and a composite cathode. A schematic representation of a lithium ion battery is depicted in Figure 1.1. During battery discharge, lithium ions move through the electrolyte from anode to cathode, and then intercalate into the oxide storage material. To preserve charge neutrality, electrons are driven through the external circuit to complete the electrochemical reaction. For proper battery operation, the electrodes must then provide fast transport for both electrons and lithium ions. Figure 1.2 summarizes the needed transport characteristics for a battery electrode.

As the intrinsic transport properties of most electrochemically active oxides are not sufficient, a three-phase porous electrode is utilized to improve the rate capability. A carbonaceous conducting additive and an electrolyte material are added to the lithium storage material in order to provide sufficient electronic and ionic conductivity, respectively. The micrograph of a composite cathode is shown in Figure 1.3. In order to optimize battery design, a clear understanding of the transport of both electrons and lithium ions in the composite electrode is necessary. Microstructural features control the critical properties in a wide variety of engineering materials, from transformation-toughened zirconia to lead ruthenate thin film resistors. Therefore, the microstructure in many materials is tailored to optimize the desired properties while reducing undesirable ones. Given that designing microstructure in a common engineer's tool, it is surprising that microstructure issues have not been fully addressed in this burgeoning field. This work considers the transport of charged species in the composite electrodes from a microstructural point of view.

Electron transport was analyzed using percolation theory and by comparison with other conductor-insulator composites. An *in situ* filter pressing apparatus was designed and constructed in order to determine the percolation threshold for electrode composite materials, and to investigate the impact of inter-particle interactions on the electronic conductivity of these composites.

Ion transport was analyzed using a computer model designed by Doyle, Fuller and Newman. Each step in lithium transport from dissolution at the cathode to intercalation and diffusion at the cathode was considered, and microstructure improvements to ameliorate the rate-limiting steps were proposed. The computer model was modified to allow the simulation of graded microstructures to evaluate the effectiveness of these improvements.

1.2 **Figures**

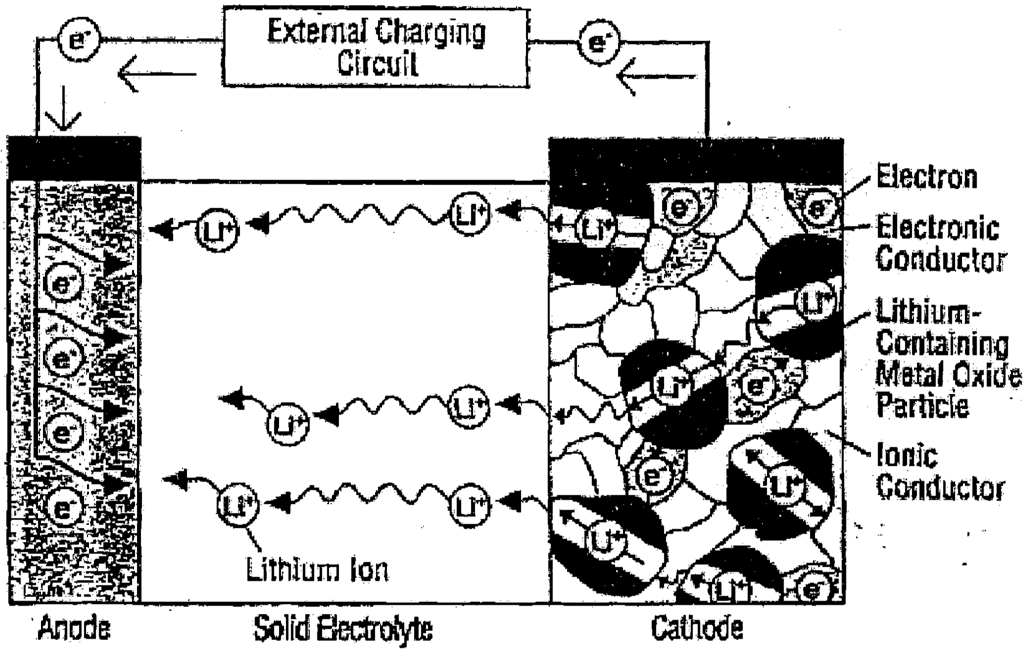

Figure 1.1 A schematic of a lithium ion battery with a composite cathode. The direction of ion and electron migration indicates that the battery is being charged. During discharge, the charged species move in the opposite direction.

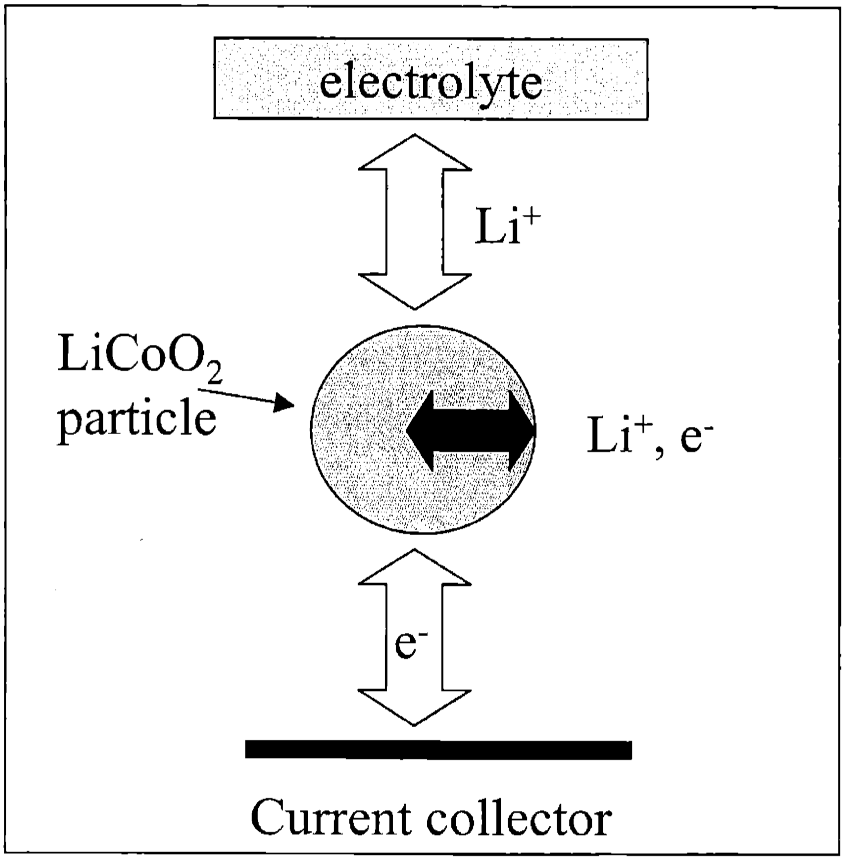

Figure 1.2 A schematic representation of the transport required in the lithium battery electrode. The electrode must provide ion paths from the intercalation oxide particle to the electrolyte separator and electron paths from the oxide particle to the current collector.

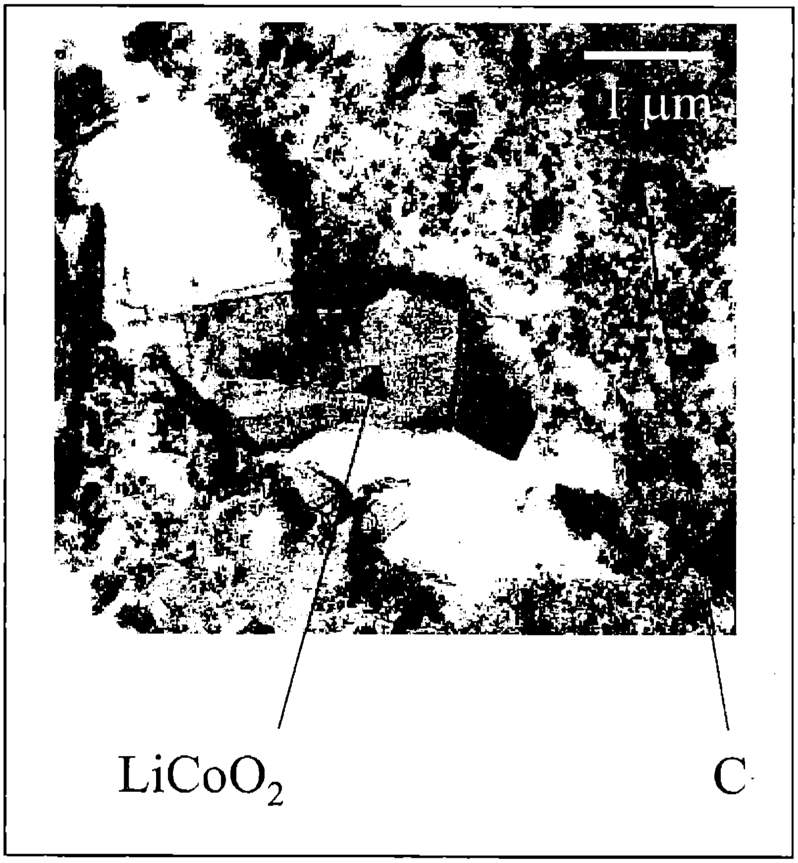

Figure 1.3 Micrograph of the composite cathode of a lithium battery. The electrode is porous which allows the liquid electrolyte to provide ion transport to the oxide particles.

# 2. Electron Transport

## 2.1 Introduction

In many conductor-insulator composites used in electronic and electrochemical devices, the dependence of electronic conductivity on phase fraction, microstructure and external conditions determine the material's usefulness. Such composites have been extensively used as thick-film resistors, thermistors, current-limiters, fuses and other applications. Similarly, mixed-phase electrode systems used as SOFC cermet anodes, or lithium ion battery electrodes rely an electronically conductive phase to provide sufficient electron transport for the electrochemical reaction to proceed. For each of these applications, controlling the electron transport properties of the composite is critical.

In order to design materials that optimize these properties, an understanding of the processes that determine electron transport is required. In conductor-insulator composites, there are two critical factors that determine the overall electronic conductivity. First, a continuous network of the conducting phase must be present in order to develop an appreciable conductivity. Percolation theory describes the presence and formation of such networks. In addition, the contact between particles must be such that it allows electron tunneling to carry current between connected particles.

Percolation theory states that the conductivity of a conductor-insulator composite will be negligible or zero until the conducting phase forms a continuous network at a volume fraction known as the percolation threshold ($p_c$). The conductivity of composites near the percolation threshold is described as a critical phenomenon, and the conductivity is power-law dependent[1].

$$\sigma \propto (p - p_c)^t \qquad [2\text{-}1]$$

where p is the conductor volume fraction, $p_c$ is the percolation threshold, and t is a non-integer positive exponent. The theoretical percolation threshold of random systems depends on dimensionality. This value has been experimentally confirmed for 3-dimensional random systems of equiaxed particles[2,3] at approximately 16 volume percent, which is in accordance with the theory. For non-equiaxed particles, the percolation threshold is reduced as the particle aspect ratio increases[4].

Percolation also depends strongly on microstructure. The percolation threshold of random dispersions is found to be higher than that of structured or agglomerated particles[5]. For example, Carcia, Ferretti and Suna [6] observed a percolation threshold as low as 2 volume percent for ruthenate-glass composites. Here smaller ruthenate grains surrounded larger glass particles; the group found that as the particle size of the ruthenate decreased, the percolation threshold also decreased. Sample preparation can also have a large effect. For example, Franco et al. showed that carbon particles in poly (ethylene oxide) percolate at 20 weight percent when prepared by a solvent casting method but at 3 weight percent when prepared with a solvent-free dry pressing procedure[7]. Again, this lowered percolation threshold can be attributed to smaller carbon particles surrounding large polymer particles in the case of the solvent free process, whereas the structure was entirely random after the polymer was cast into solution.

The inter-particle conduction mechanisms in conductor-insulator composites have been extensively studied due to their varied device applications[3,8-10]. The systems that have received the most attention are cermet thick-film resistors[8] and polymer-carbon black composites with a positive temperature coefficient of resistance (PTCR) effect[9]. In both of these materials, the conductivity is limited by tunneling across conductor particle contacts[8]. In cermet thick-film resistors, this tunneling current is controlled by the thickness and composition of a thin intergranular glass film[11]. Impurities within the tunnel barriers act as resonant centers and enhance tunneling[8].

In polymer-carbon black composites, tunneling between carbon particles likewise forms the conduction path. At temperatures approaching room temperature, these composites instead exhibit a PTCR, as the resistance increases with increasing temperatures. As the polymer matrix is heated through the melting point, the material exhibits a PTCR anomaly as the resistivity increases by several orders of magnitude over a temperature range of only a few degrees K. Usually it is assumed that an increase in tunneling length that occurs at the melting point due to volume of the polymer matrix is the cause of this anomaly[9], as the tunneling conductance decreases exponentially with increasing particle separation[11]. Alternately, this effect is attributed to the making and breaking of carbon-carbon contacts as the matrix expands[2]. Heaney has studied critical behavior in these systems, and concluded that percolation effects alone cannot account for the resistivity change with temperature[3].

Lithium battery electrodes are also processed as suspensions of carbon and oxide particles in an electrolyte. As the intrinsic electronic and lithium conductivities of cathode intercalation compounds such as $LiCoO_2$ and $LiMn_2O_4$ and carbon-based anode materials are too low for them to function as dense bulk materials, a porous composite electrode has been used in most battery designs. Transport into and out of the oxide particles is dependent upon additives to this composite electrode. A liquid electrolyte provides ion transport through the pore network, while a carbon additive provides electronic conductivity. While continuous conductor and electrolyte networks are necessary for adequate global ionic and electronic transport in the electrode, the highest possible fraction of the lithium storage material is desired in order to maximize energy density.

Clearly, both the percolative aspects of the microstructure, and the local inter-particle tunneling properties control the electronic conductivity of all of these multi-phase composites. Both of these factors are strongly affected by the morphology and intrinsic properties of the starting materials as well as the inter-particle interactions. Inter-particle interactions have been extensively studied in colloidal suspensions in order to develop effective means of processing them. In colloidal suspensions, three types of surface forces are commonly observed.

The van der Waals interaction is always attractive for like particles in any medium, and it will dominate the inter-particle interaction at short distances, leading to good inter-particle contacts. For spherical particles at small separations, the van der Waals potential energy is inversely proportional to the distance between particles[12]. This interaction leads to a primary minimum in interaction energy at very close particle approach.

In electrolytic suspensions, a Derjaguin, Landau, Verwey & Overbeek (DLVO) double layer may also form. This occurs when ions in the solution adsorb onto and charge particle surfaces. Formation of such a double layer can lead to electrostatic repulsion that stabilizes the suspension. This effect would hinder close particle approach, and lead to poor inter-particle contacts[13]. Double layer stabilization occurs when the double layer is sufficiently diffuse that it provides an energy barrier to particle approach. Typically, DLVO interactions stabilize suspensions at salt concentrations below 0.1M if the ions are monovalent. At higher concentrations, the double layer is compressed, and van der Waals forces predominate, leading to particle aggregation[13].

Finally, a suspension may be stabilized sterically. In ceramics processing, surfactant molecules are commonly added to particle suspensions in order to control inter-particle interactions. Usually, these surfactants have a polar and non-polar end, or else an ionically charged and a neutral end. Selective adsorption of one end of the molecule to the particles prevents contact formation because the surfactant molecules are interposed between the particles[13]. The steric repulsion is dependent on the thickness of the adsorbed layer and on the chemical nature of the adsorption.

In cermet thick-film resistors, the inter-particle forces have been studied by Chiang et al.[11] They found that van der Waals attractions are sufficient to keep the ruthenate particles flocced, but that a nanometer thick glass film provides steric repulsion between individual grains at equilibrium. The group also demonstrated that titania addition increased the film thickness, and they were able to use this effect to control the resistivity and the TCR for these materials.

In this work, an *in situ* filter pressing apparatus was designed and constructed in order to allow the convenient measurement of electronic conductivity in a solid-liquid composite under changing solid loading. Using this apparatus, it was also possible to qualitatively observe the effect of inter-particle interactions on the conductivity of a conductor-insulator composite.

## 2.2 Experimental

Apparatus

An *in situ* filter pressing apparatus (Fig. 2.1) was constructed to study the effect of inter-particle forces and percolation on the electronic conductivity of lithium battery cathode composite materials. Quartz tubes, custom etched to an inner diameter of ¼ in. (G. Finkenbeiner, Waltham, MA) were used as the sleeve for the filter-pressing die. Use of a transparent sleeve allowed the direct measurement of the column height of the composite. In addition, it allowed visual inspection of the quality of the seal as well as the uniformity of the composite mixture within the press. The pistons were made of ¼ inch OD steel tubes. A 1/8-inch hole was drilled down the center of each piston in order to allow liquid to expire from the composite mixture. These hollow steel pistons were then fitted with porous metal disks (Mott Corp., Farmington, CT). The disks had an outer diameter of ¼ in., thickness of 0.062 in. and an average pore size of 0.5 μm. These disks were attached to the pistons using silver-loaded conductive epoxy (E-Solder No. 3012 VonRoll Isola, Liverpool, NY) to ensure both good adhesion as well as electrical contact. The epoxy was cured at 150 °C for 4 hours. Teflon tape was used to ensure a good seal between the pistons and sleeve, and to aid in visual monitoring of the pressing process. Leads were attached to each piston to permit the resistance measurements.

In the experimental measurements, the sample resistances ranged from several ohms for compacted systems to several mega ohms for systems with loading below the percolation threshold. Because the geometric configuration of the pressing apparatus precluded four-point probe electrical measurements, control measurements were performed in order to ensure the validity of two-point probe measurements. Resistance at contact between the two pistons in the absence of samples was less than 1 Ω. Since graphite was the conductive phase tested in the experiments, the press was then loaded with graphite powder and highly compacted to 63% of theoretical density. The resistance was again less than 1 Ω. These results showed that contact resistance between the metal filters and the test material was insignificant at high compaction. As a result, contact resistances could be neglected, and the two-point probe resistance measurements were considered indicative of the true sample resistance.

Each suspension was prepared for testing by mechanically mixing the solid particles with the chosen liquid medium. After mechanically mixing and stirring for 5 minutes to ensure uniformity, the solid particles were ultrasonically dispersed for 15 minutes. Once ultrasonically dispersed, the suspensions were loaded into the pressing apparatus and compressed. Upon pressing the samples, the liquid discharge from the hollow pistons was clear and free of any particles, indicating a good seal both between the piston and the glass sleeve, as well a sufficiently fine pore size in the metal filters to prevent solid penetration. The height of the suspension column was precisely measured, and the resistance was simultaneously recorded during pressing. From these data, the volume fraction of solids and the resistivity of the cake were calculated. The volume fraction is given by:

$$V_f(C) = V_f^0(C)\frac{h}{h^0} \qquad [2\text{-}2]$$

where h is the column height, $h^0$ is the initial column height, and $V_f^0(C)$ is the initial volume fraction of graphite. The resistivity is given by:

$$\rho_e = R_{sample}\frac{A}{h} \qquad [2\text{-}3]$$

where $R_{sample}$ is the measured resistance and A is the cross-sectional area of the pressing die.

## Materials Systems

Measurements were performed on suspensions of graphite and doped $LiCoO_2$ in various liquids representative of lithium ion battery electrolytes. All experiments were conducted in an argon-filled dry box.

Graphitic carbon, 1-2 μm particle size (Aldrich, Milwaukee, WI) was used as the conductive phase in most of the experiments. Mg-doped $LiCoO_2$ was used in experiments testing the composite conductivity of cathode oxides. This composition was used rather than pure $LiCoO_2$ due to its improved electronic conductivity[14]. The oxide with stoichiometry $LiMg_{0.05}Co_{0.95}O_2$ was prepared by mixing of stoichiometric amounts of $Co(OH)_2$ and $Mg(OH)_2$ with $Li_2CO_3$ (Aldrich, Milwaukee, WI), and firing in air at 800°C for 6 hours. This powder was characterized by X-ray diffraction (XRD), and confirmed to have the α-$NaFeO_2$ structure type. TEM photographs of both graphite and $LiMg_{0.05}Co_{0.95}O_2$ particles were obtained. The graphite particles appeared to range in size from 0.5 to 1.5 microns, and exhibited a hexagonal morphology. The TEM Bright Field Image of typical graphite particle is shown in Figure 2.2. Figure 2.3 shows a small cluster of graphite particles. The $LiMg_{0.05}Co_{0.95}O_2$ particles exhibited a hexagonal morphology similar to the graphite particles. The oxide particles were larger than the graphite particles, with a size range of 1-3 microns for each oxide particle. A TEM image of a typical oxide particle is shown in Figure 2.4. A cluster of oxide particles is shown in Figure 2.5.

Anhydrous ethylene carbonate (EC), dimethyl carbonate (DMC), diethyl carbonate (DEC) and 1,2,dimethoxyethane (DME) electrolytes (Alfa Aesar, Word Hill, MA) were used. $LiPF_6$ salt (Alfa Aesar, Word Hill, MA) was added to the pure solvents. A salt concentration of 1M, typical for liquid electrolytes, was used for most tests. Solutions of 1:1 v/v EC/DMC electrolyte with salt concentrations of 0.1M and 0.01M were prepared.

A solution of a novel block co-polymer electrolyte (BCE) in anhydrous THF (Aldrich, Milwaukee, WI) was also used. The BCE used was a tri-block co-polymer consisting of 17% PLMA-66% POEM-17%PLMA, with a molecular weight of 70000, prepared by P. Soo[15]. The solution contained 5 percent polymer by weight. The polymer was doped with $LiCF_3SO_3$ (Aldrich, Milwaukee, WI) at a ratio of 20:1 [EO:$Li^+$]. To compare this BCE to a typical polymer electrolyte, tests were also performed on poly (ethylene oxide) (PEO). PEO (100K Molecular weight, Poly Sciences Warrington, PA) was added to THF at 5 percent by weight. At room temperature, the PEO did not dissolve in the THF. Tests were subsequently performed on suspensions where the PEO was not dissolved (suspensions of carbon *and* PEO particles in THF) and also on suspensions where the PEO was dissolved in THF by heating to 90 °C for 15 minutes and allowing the PEO to dissolve. Upon subsequent cooling, the PEO remained in solution. A control experiment with carbon suspensions in pure THF was also performed.

## 2.3 Results and Discussion

Figure 2.6 shows the effect of the liquid electrolyte on the resistivity of graphite suspensions as a function of volume fraction. At a volume fraction of 10% graphite, the resistivity of each suspension drops by two orders of magnitude from ~$10^5$ to ~$10^3$ Ω·cm. It appears that percolation of graphite particles occurs at volume fractions between 10 and 13% rather than the 16% expected for random bond percolation. Agglomeration of graphite particles due to van der Waals forces could explain this effect. However, the TEM images show that both the carbon and oxide particles were readily dispersed. It is therefore more likely that the non-spherical morphology of the particles reduced the percolation threshold from 16% to the observed value. As the volume fraction of carbon increases further, the resistivity slowly decreases to between 10 and 100 Ω·cm at 40 volume percent carbon. Graphite has an intrinsic resistivity on the order of $10^{-3}$ Ω·cm. This large difference between the intrinsic resistivity of graphite and the resistivity measured in the tests indicates that it is the inter-particle contacts, rather than the bulk resistivity of the particles that determines the overall electrical properties of the suspension.

The various liquid electrolytes showed similar composite behavior at low volume fractions. At solid volume fractions near 0.3, the sample pressed in DEC has a higher resistivity. Of the liquid electrolytes tested, DEC has the highest molecular weight; it is possible that the polar carbonate group of the DEC selectively adsorbed on the carbon surfaces, and the aliphatic ethyl groups at either end provided some resistance to particle approach. At low graphite loadings, the DEC also seems to exhibit a percolation threshold closer to the theoretical value than the other liquids, which may mean that the graphite is better dispersed in this medium.

The effect of salt concentration on the resistivity of graphite suspensions in EC/DMC is shown in Figure 2.7. At all measured salt concentrations, the percolation threshold again occurred at 10 volume percent. The resistivity of the suspension decreased from $10^5$ to $10^3$ $\Omega$·cm at this volume fraction. As the graphite loading increased from this level to 30 vol. %, the resistivity again decreased slowly to a level between 10 and 100 $\Omega$·cm. It appears that salt concentration has little effect on the resistivity of the suspension, suggesting that DLVO interactions do not have a significant effect on graphite particle aggregation in the present liquid battery electrolytes, as both high- and low-salt solvents show electronic percolation at nearly identical carbon loadings. Typically, DLVO interactions stabilize aqueous suspensions at salt concentrations less than 0.1 M for electrolytes with monovalent salts. In the present case, both the pure medium and also a 0.01M $LiPF_6$ solution in an EC/DMC electrolyte behave almost identically to the electrolytes with higher salt concentrations.

Figure 2.8 shows that a polymer solution dramatically changes the electrical properties of the graphite-electrolyte suspension. Here, a suspension of carbon in the block copolymer electrolyte solution is compared to carbon powder compacted in air and to a suspension in EC/DMC liquid electrolyte. The resistivity of graphite powder compacted in air shows very similar behavior to that of the carbon-EC/DMC suspensions, indicating that the liquid electrolyte does not interfere with the formation of inter-particle contacts. However, in the BCE-electrolyte solution, the apparent percolation threshold is shifted to a much higher loading of 25 vol.%. At 10% graphite loading, where suspensions in liquids and air have already percolated, the suspension in polymer solution has a resistivity that is approximately 5 orders of magnitude greater. At the theoretical percolation limit of 16%, the resistivity remains 3 orders of magnitude greater. Thus, it can be concluded that the polymer is acting as a surfactant in this solution, interfering with the formation of electronically conductive carbon-carbon contacts. At 45 percent carbon loading, however, the resistivity converges with that found in graphite-liquid electrolyte suspensions, indicating that even modest pressure is able to force the carbon particles into contact despite the adsorbed polymer.

The resistivity of $LiMg_{0.05}Co_{0.95}O_2$-electrolyte suspensions is shown in Figure 2.9. Again, it appears that oxide particles in liquid electrolytes are able to form good contacts, indicated by the sudden decrease in resistivity from $10^5$ to $10^3$ $\Omega$·cm as the carbon loading exceeds 10 volume percent. This resistivity remains almost constant as the suspension is further compacted to a solid loading of 50 volume percent. The higher resistivity upon further compaction, compared with the carbon suspensions, may be due to the higher intrinsic resistivity of the doped oxide (~1 $\Omega$·cm)[14]. However, the solid polymer electrolyte solution here also appears to impede the formation of electronically conducting contacts between oxide particles, as the resistivity is greater across most of the volume fraction range. Only at very high particle loadings does the resistivity begin to approach that of the oxide suspensions in liquid electrolytes. Even at 50 percent oxide loading, the suspension in the polymer solution has a resistivity that is an order of magnitude higher than that of the oxide suspension in liquid electrolyte. This result indicates that the solid polymer electrolyte also acts as a surfactant on the oxide particles, interfering with electronic percolation over a broad range of volume fractions.

In Figure 2.10, the resistance is plotted against graphite volume fraction for suspensions in a PEO and THF solution. Two different states of PEO in THF have been tested. The samples in which the PEO had been dissolved by heating to 90 °C showed behavior similar to that of BCE solutions in THF; it appears that the polymer solution interferes with the electrical contact between graphite particles. On the other hand, the samples in which the PEO was suspended in particulate form, but not dissolved showed behavior very similar to that of a control suspension of THF with no polymer. It appears that in particulate form, the polymer does not interfere with inter-particle contacts. This conclusion is consistent with previous studies on polymer-graphite composites, where it was found that solution free mixtures percolated at much lower graphite loadings than samples prepared with polymers in solution[7].

Figure 2.11 compares the behavior of graphite suspensions in PEO solutions to graphite suspensions in BCE solutions. The dissolved PEO has a similar effect on carbon-carbon contacts in a suspension as the dissolved BCE. In both cases, the polymer is found to interfere with the formation of conductive carbon-carbon contacts when compared with the control samples of graphite suspended in pure THF.

The surfactant effect indicates that it may be necessary to increase carbon loading in order to ensure sufficient electronic conductivity in a solid polymer battery. Furthermore, due to lower ionic conductivity compared with liquid electrolytes, it is to be expected that a greater fraction of solid electrolyte is necessary to ensure sufficient ion transport.

## 2.4 Conclusions

An *in situ* measurement apparatus was built in order to probe the electronic conductivity as a function of composition in lithium battery mixed-phase electrodes. This apparatus worked by means of a filter pressing mechanism to allow the measurement of sample resistivity while changing the volume fraction of conducting phase. A broad range of phase fractions, most notably spanning the percolation threshold was characterized. Because the geometric configuration of the apparatus precluded four-point probe measurements, the validity of two-point probe electrical tests was experimentally confirmed.

The percolation limit for graphitic carbon and lithium cobalt oxide dispersed in organic liquid electrolytes occurs between 10 and 13 vol. %. This lowering of the percolation threshold compared to the theoretical 16 vol. % value can be ascribed either to anisotropic particle morphology or to van der Waals interactions that encourage the formation of electronically conducting contacts in a looser packing arrangement. Neither the choice of organic electrolyte nor the salt concentration has a significant effect on electronic conductivity controlled by inter-particle contacts. On the other hand, polymer electrolytes in solution cause an increase in the volume percolation limit to about 25% and a decrease in the electronic conductivity at low solid volume fraction. This phenomenon is attributed to polymer surfactant adsorption.

An advantage of the *in situ* measurement apparatus is that it allows continuous measurements on a single specimen, yielding data that would otherwise have required the preparation of a series of samples. Due to the low strength of the quartz sleeves used in this instance, the apparatus was not suitable for use under high pressure. Further modifications may allow high-pressure measurements. Nonetheless, this measurement technique has provided valuable information about electronic conduction in mixed phase lithium battery electrodes and has promise as an investigative tool for other conductor-insulator composites.

### 2.5 Figures

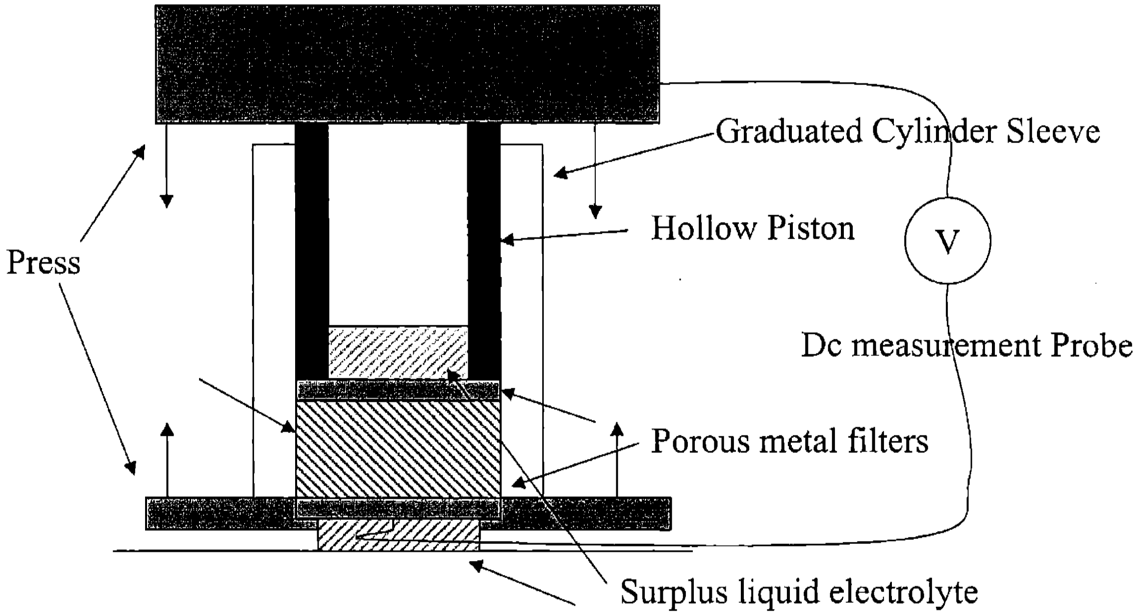

Figure 2.1 Schematic representation of the *in situ* filter pressing apparatus

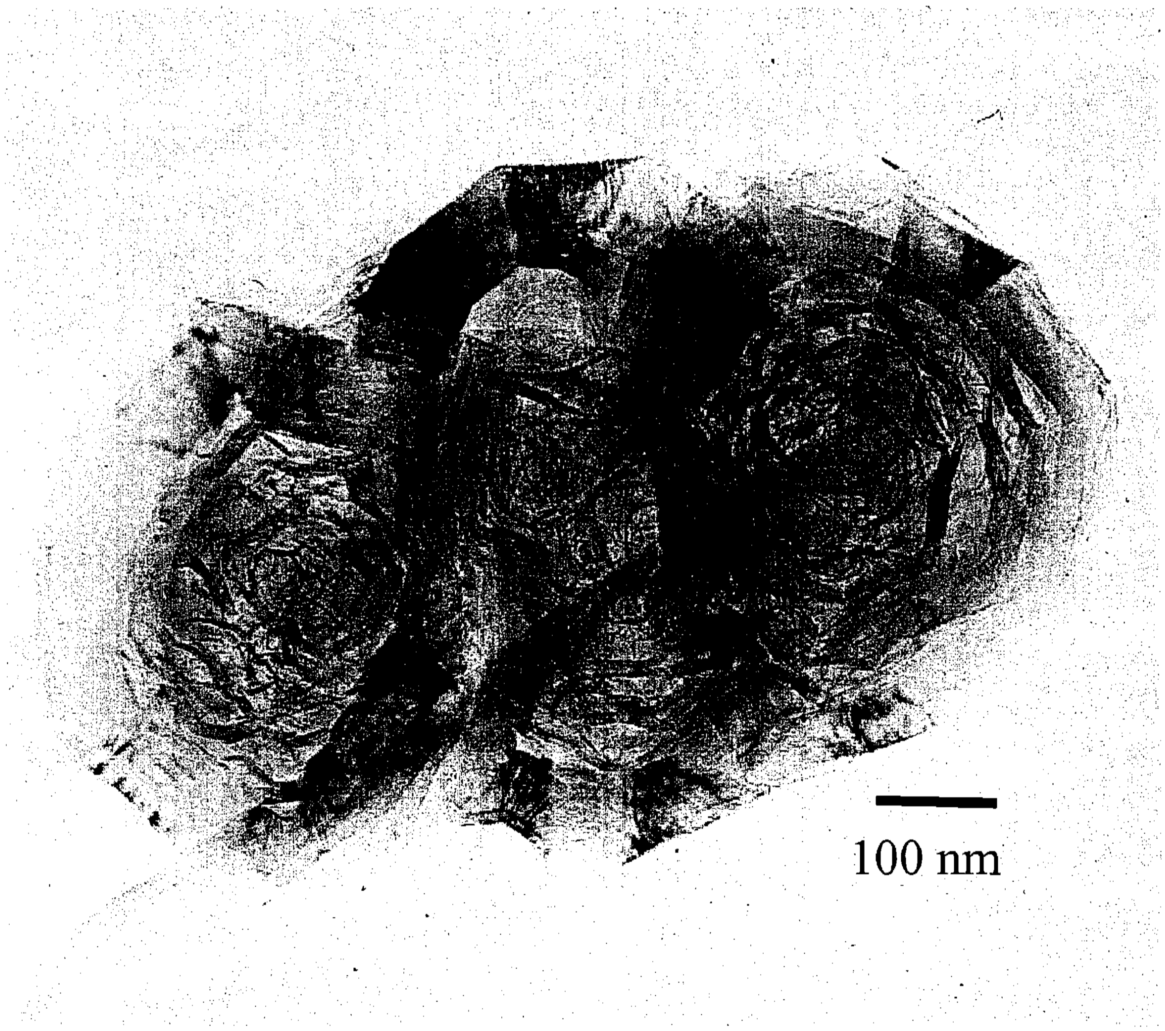

Figure 2.2 TEM Bright Field Image of a typical graphite particle used to prepare the suspensions at 85K magnification.

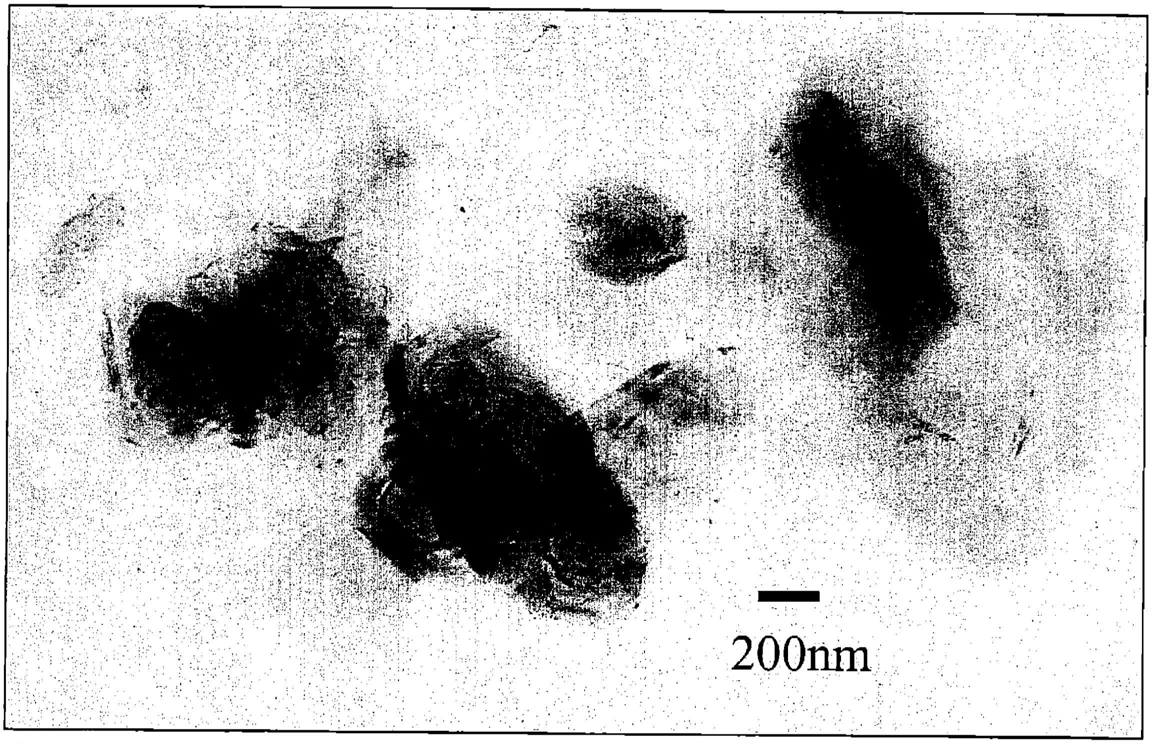

Figure 2.3 TEM Micrograph of a cluster of graphite particles at 25K magnification

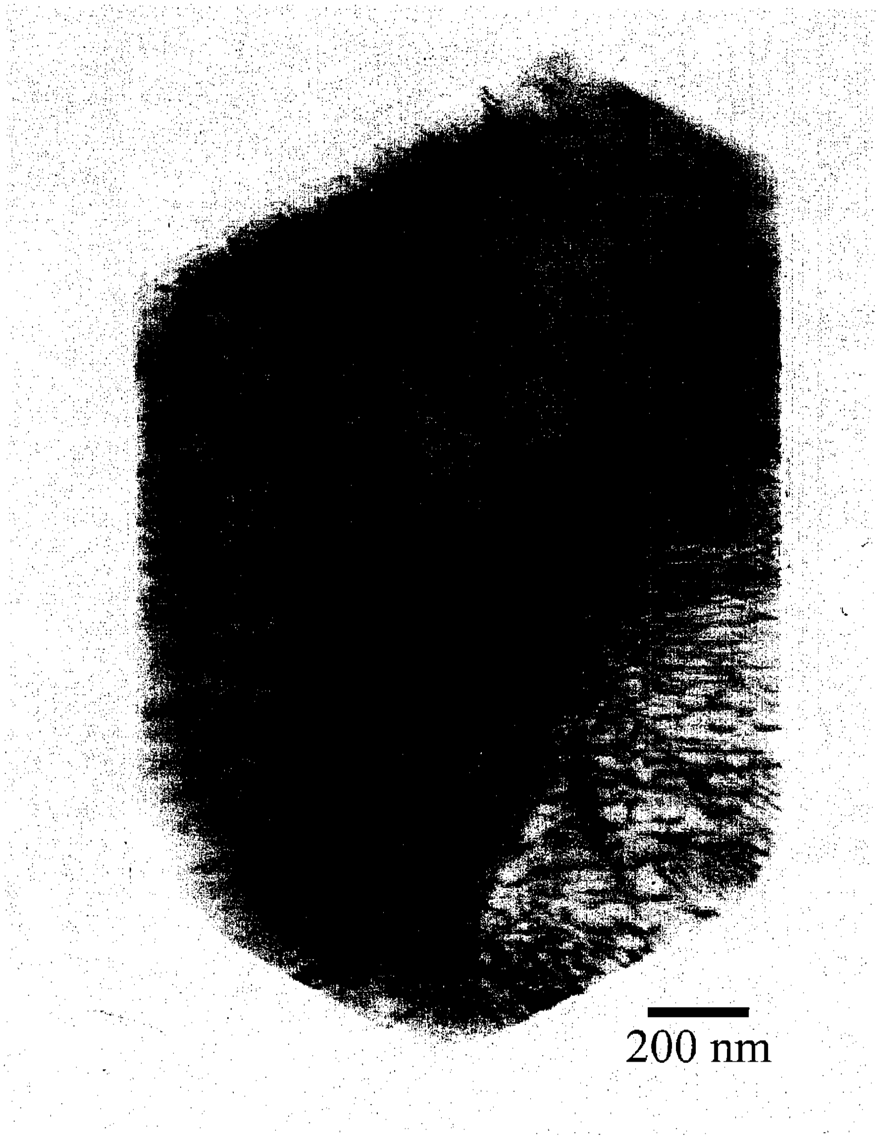

Figure 2.4 TEM Image of a typical $LiMg_{.05}Co_{.95}O_2$ particle used to prepare the suspensions at 34K magnification.

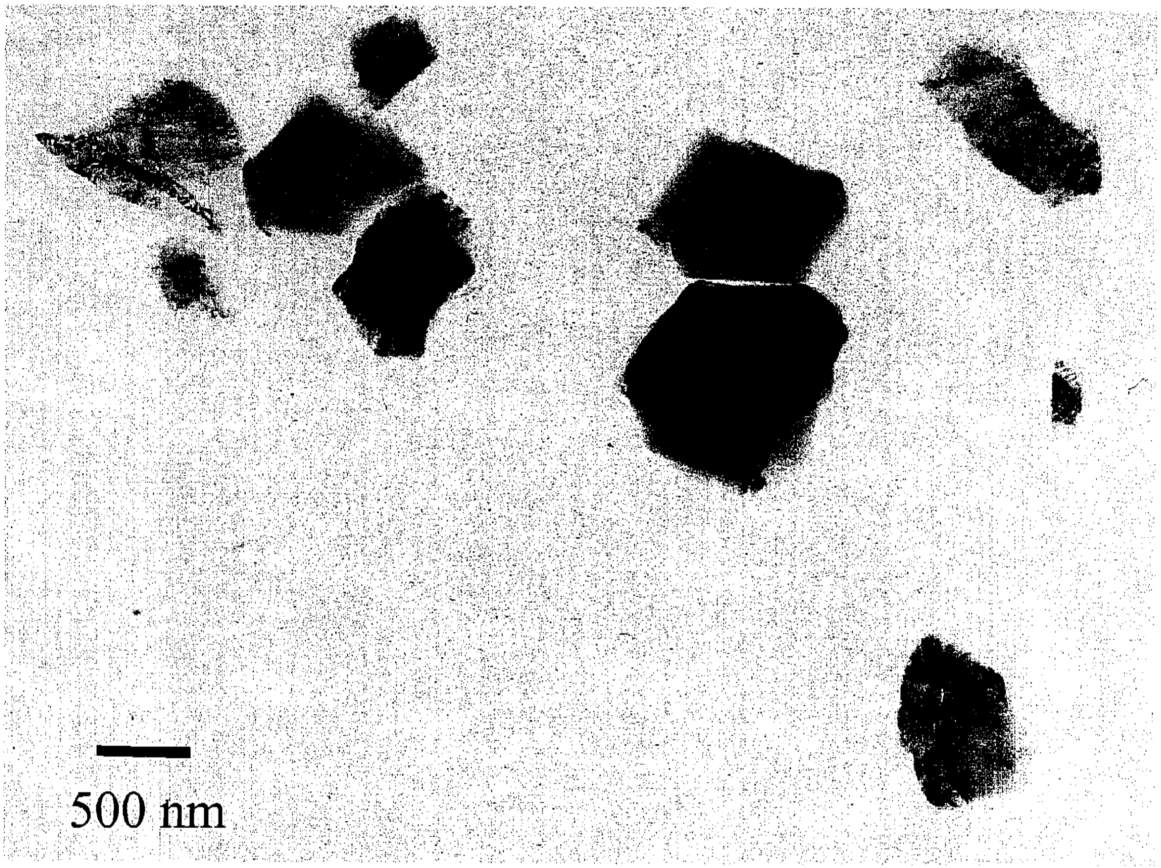

Figure 2.5 TEM Bright Field Image of a cluster of $LiMg_{.05}Co_{.95}O_2$ particles at 17K magnification.

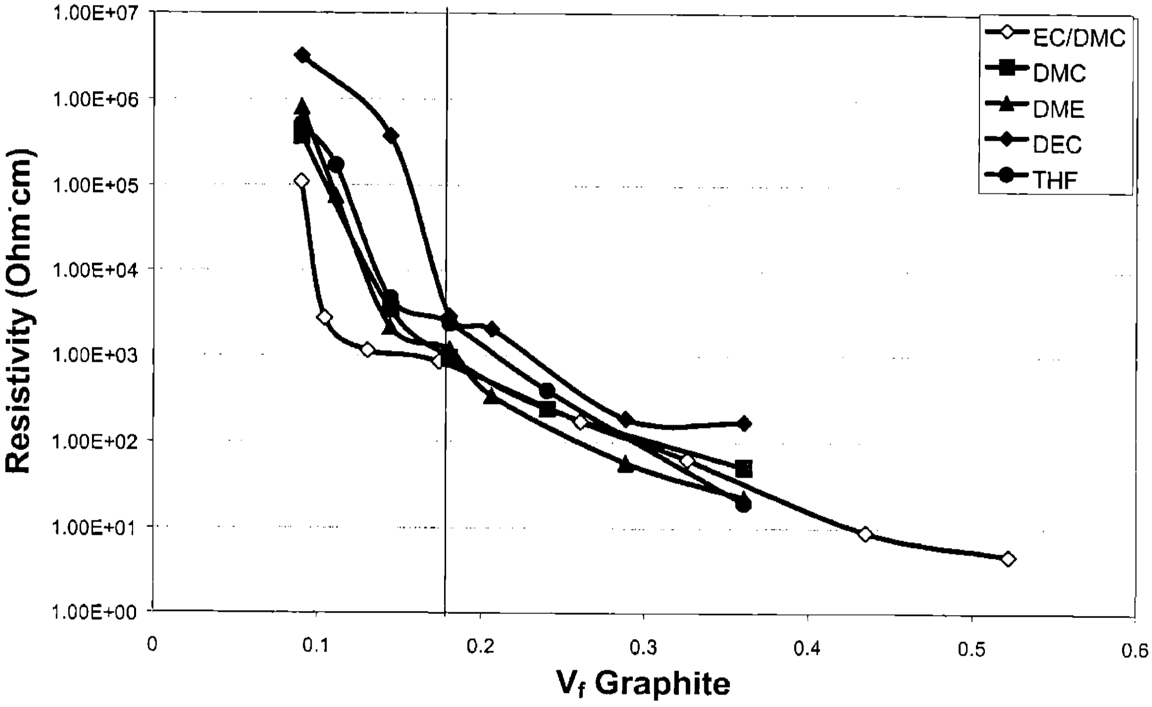

Figure 2.6 Resistivity against volume fraction for graphite in various liquid electrolytes with 1M $LiPF_6$. The vertical line represents the bond percolation threshold for randomly packed spheres. The suspensions exhibit electronic percolation at ~10 vol.% carbon.

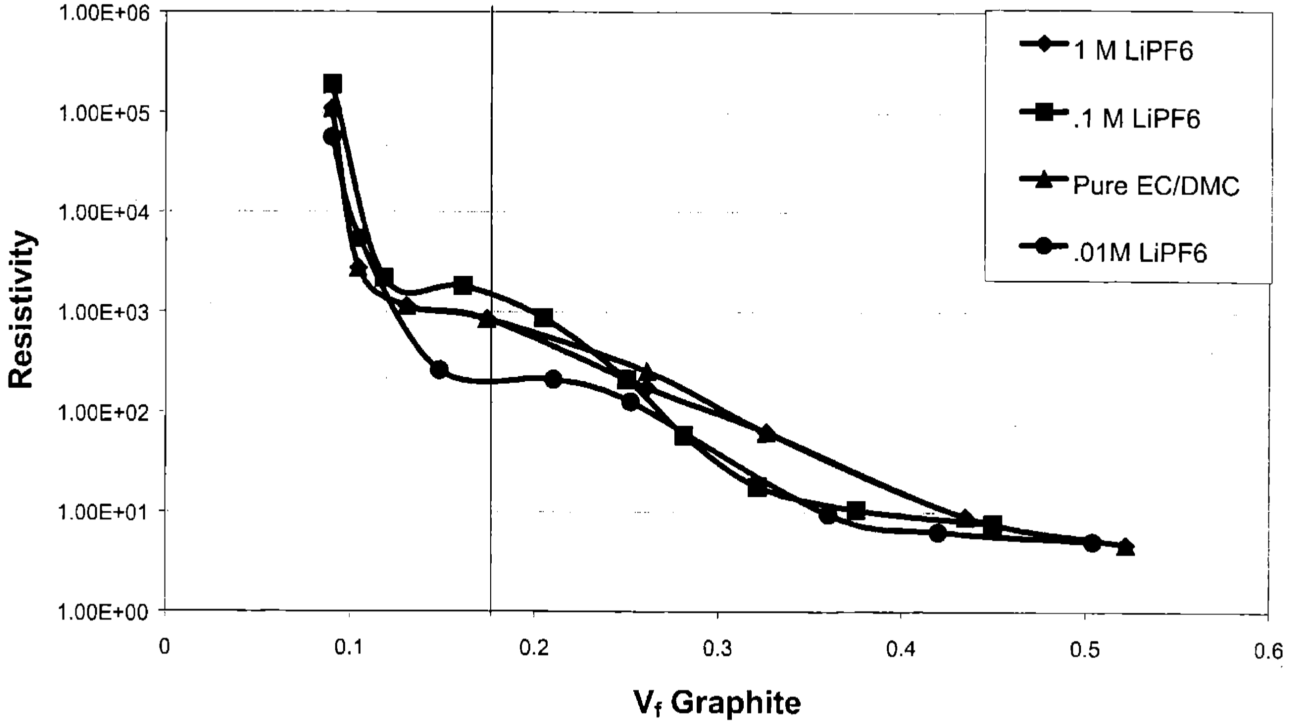

Figure 2.7 Resistivity against volume fraction for graphite in EC/DMC electrolyte with varying $LiPF_6$ salt concentration. The vertical line represents the bond percolation threshold for randomly packed spheres. The suspensions exhibit electronic percolation at ~10 vol.%.

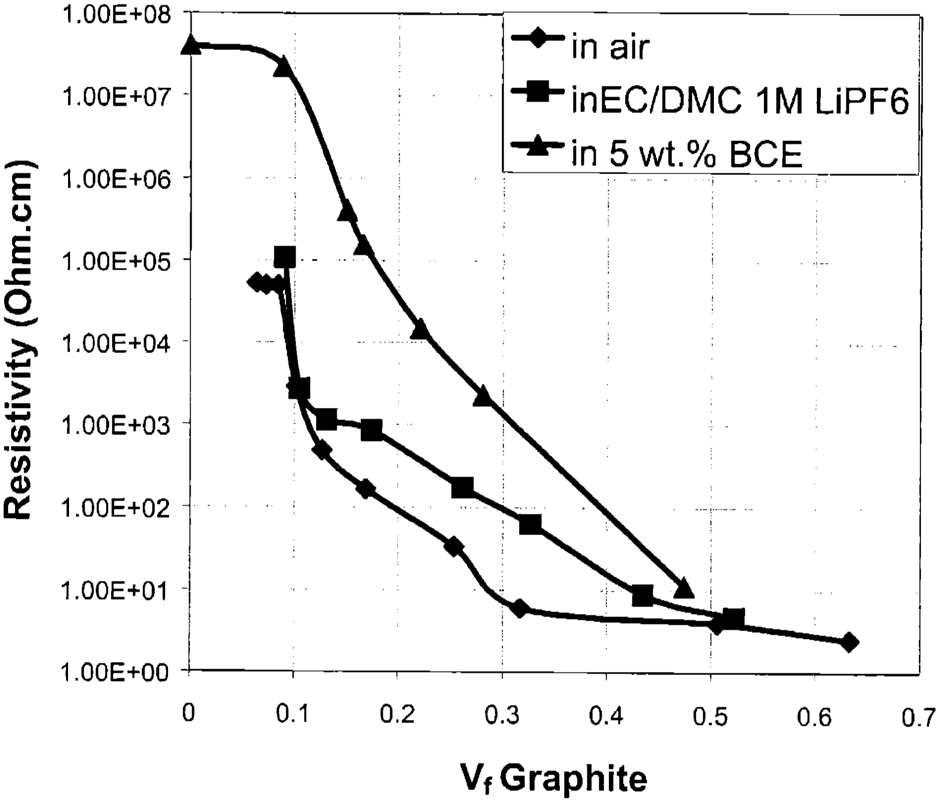

Figure 2.8 Resistivity against volume fraction for graphite in liquid electrolyte vs. solid polymer electrolyte solution. The liquid electrolyte is EC/DMC 1M $LiPF_6$. The polymer solution is a 5 w/o PLMA-POEM-PLMA triblock copolymer in THF doped with $LiCF_3SO_3$. The polymer solution interferes with electronic percolation at low graphite loadings, exhibiting a resistivity that is 5 orders of magnitude greater at 10 vol.% carbon loading.

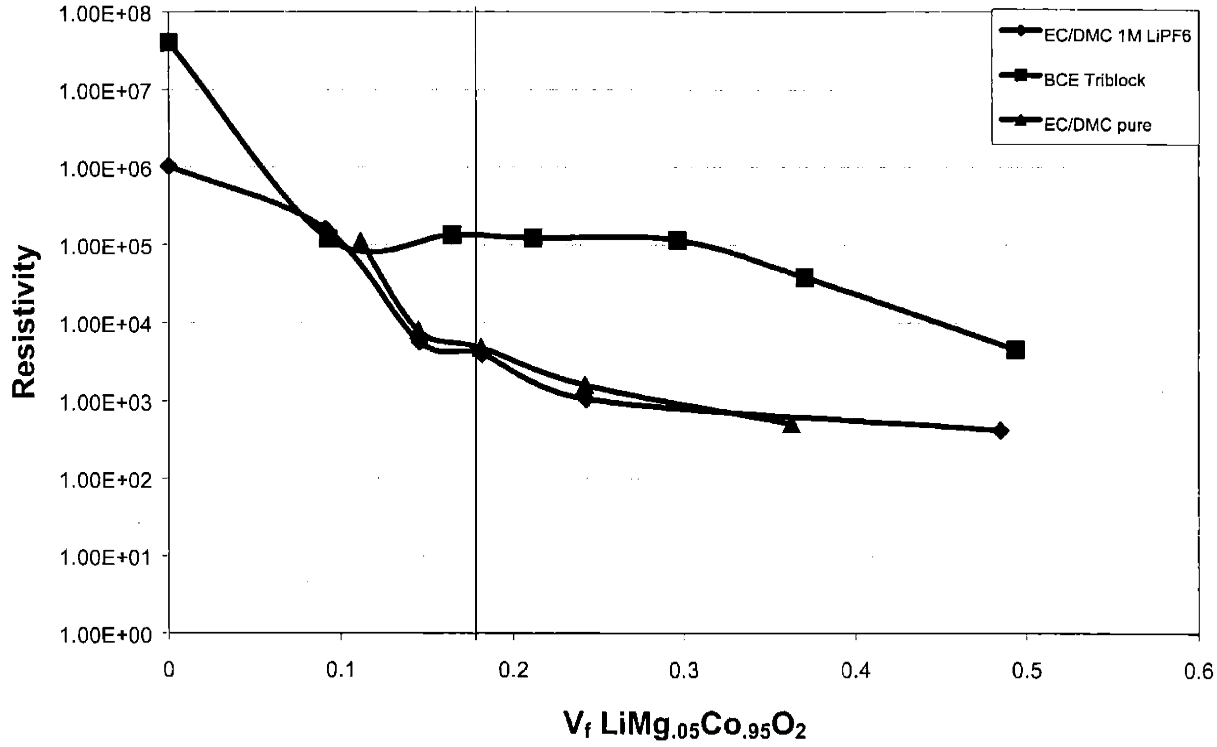

Figure 2.9 Resistivity against volume fraction of $LiMg_{0.05}Co_{0.95}O_2$ in various electrolyte media. Again, the polymer solution is observed to interfere with electronic percolation in the suspension. The decrease in resistivity with increasing solid volume fraction is not as dramatic as in carbon suspensions, due to the higher intrinsic resistivity of the doped oxide when compared with carbon.

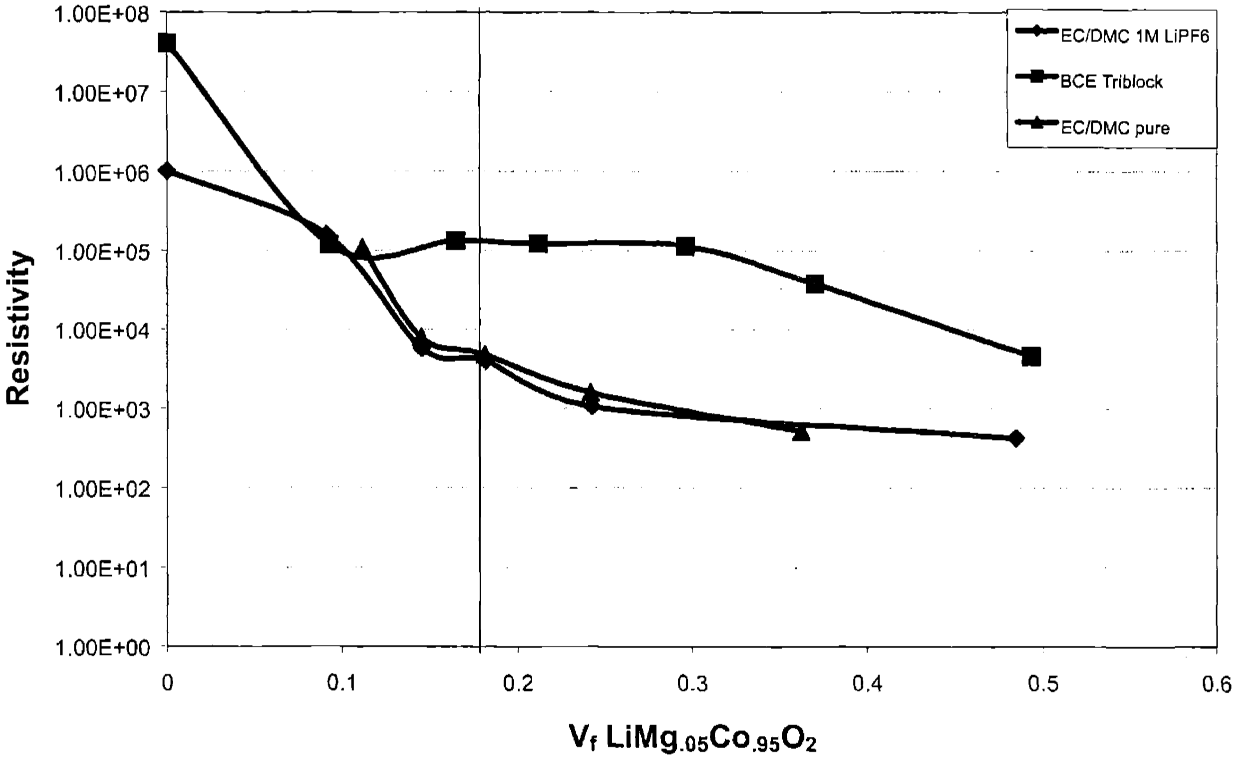

Figure 2.10 Resistivity against volume fraction of Graphite in PEO/THF. The PEO solution is observed to interfere with electronic percolation in the suspension, whereas the suspension of PEO particles in THF shows behavior similar to the control experiment with carbon suspended in pure THF. The vertical line indicates the 16 vol. Percent theoretical percolation threshold.

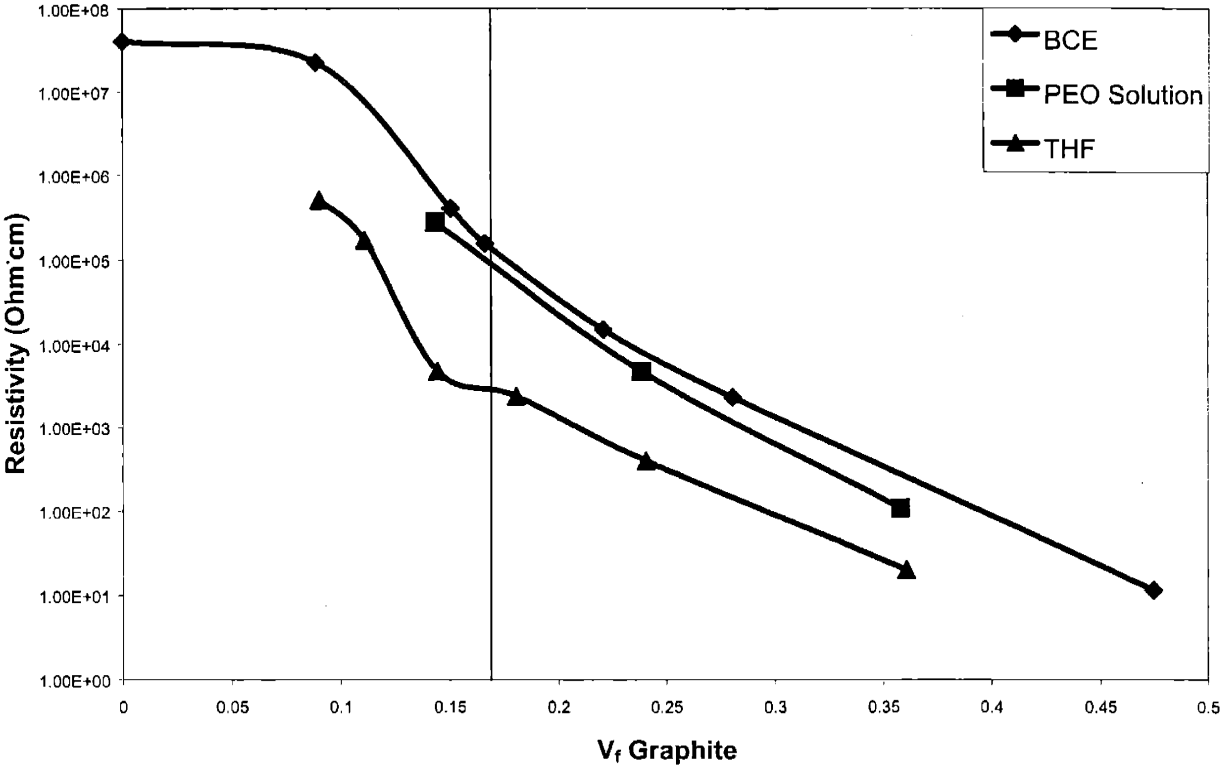

Figure 2.11 Resistivity against volume fraction of Graphite in polymer solutions. Both the BCE and the PEO solution are observed to interfere with electronic percolation in the suspension, whereas the control experiment with carbon suspended in pure THF shows percolation behavior similar to that of other liquid electrolytes. The vertical line indicates the 16 vol. Percent theoretical percolation threshold.

# 3. Ion Transport

## 3.1 Introduction

During battery operation, lithium ions must pass through several steps in order to complete the electrochemical reaction. These steps consist of:

1. Dissolution of lithium at the anode surface, freeing an electron for the external circuit
2. Transport of lithium ions across the electrolyte separator
3. Transport of lithium ions through the electrolyte phase in the composite cathode
4. Intercalation into the active cathode material, consuming electrons from the external circuit
5. Solid state diffusion of lithium into the active material, and electron transport from the current collector to the intercalation sites

In considering methods to accelerate lithium transport through microstructure design, it is first necessary to determine which of these steps is rate-limiting. Once the limiting step has been identified, it may be possible to design microstructures that will accelerate it.

Both lithium dissolution at the anode surface and the intercalation reaction at the cathode-electrolyte interface are thermally activated, and are described by Butler-Volmer reaction kinetics, where the reaction transfer current is given by[16]:

$$i_{xfr} = i_o\left[\exp\left[\frac{\alpha_{a1} F \eta_{s1}}{RT}\right] - \exp\left[\frac{\alpha_{c1} F \eta_{s1}}{RT}\right]\right] \qquad [3\text{-}1]$$

where $i_0$ is the exchange current density, $\alpha_{a1}$ and $\alpha_{c1}$ are symmetry factors for the anodic and cathodic reaction, $\eta_{s1}$ is the potential difference between the two phases at the reaction site, and *F* is the Faraday constant. The charge transfer reactions at both electrodes are thought to be fast at room temperature, and thus are not rate limiting[17]. In situations where this reaction does become rate-limiting, the charge transfer reaction can be accelerated by increasing the surface area of the reaction. This can be done by reducing the particle size of the intercalation material.

Lithium intercalation into the electrode particles is driven by solid-state diffusion, which is governed by Fick's law:

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x}\left[ D \frac{\partial c}{\partial x} \right] \qquad [3\text{-}2]$$

The time t required for a species to diffuse a distance x is approximated by

$$t = \frac{x^2}{D} \qquad [3\text{-}3]$$

For the typical intercalation oxides, D is approximately $10^{-9}$ $cm^2/s$ at room temperature[18,19] and the diffusion distance is on the order of 1 μm, giving approximately 10 seconds as the diffusion time. A battery is usually cycled over a time on the order of hours, indicating that solid-state diffusion does not limit battery operation. Diffusion limitations are also easily removed by reducing the oxide particle size.

Ion transport in the liquid electrolyte phase occurs in two separate regions. In the separator region, no electrochemical reaction takes place, and transport is governed entirely by the intrinsic transport properties of the electrolyte. This region gives rise to ohmic losses which can only be reduced by improving the intrinsic transport properties of the electrolyte, or by reducing the separator thickness.

In the composite electrode, ion transport occurs through electrolyte-filled pore channels. The ion transport still occurs in the electrolyte phase; however, the lithium transport properties must be adjusted for the tortuosity of the ion path through the composite electrode. In addition, the ionic current changes with depth in the electrode due to the electrochemical reaction. In the liquid phase of the composite electrode, the lithium material balance is expressed by[16]:

$$\varepsilon \frac{\partial c}{\partial t} = \nabla \cdot \varepsilon D_{eff} \nabla c + a j_n \left(1 - t_+^0\right) \qquad [3\text{-}4]$$

where ε is the volume fraction of electrolyte, $D_{eff}$ is the effective diffusion coefficient, $t_+^0$ is the lithium transfer number, *a* is the particle surface area available for the reaction and $j_n$ represents the local lithium flux density across the electrolyte-particle interface due to the intercalation reaction. The electrical potential gradient in the liquid phase will be[16]:

$$\nabla \eta = \frac{i_{ion}}{\kappa_{eff}} + \frac{RT}{F}\left(1 + t_+^0\right) \nabla \ln c \qquad [3\text{-}5]$$

where $\kappa_{eff}$ is the effective ionic conductivity in the solution phase, and $i_{ion}$ is the ionic current in the solution phase. Finally, the ionic conductivity, κ and the diffusivity, D in the solution phase must be adjusted to their effective values given the porosity. For a conducting matrix containing non-conducting spheres, the Bruggeman correction applies [20]. The effective ionic conductivity and diffusivity will be given by

$$\kappa_{eff} = \varepsilon^{3/2} \kappa_0 \qquad [3\text{-}6]$$

$$D_{eff} = \varepsilon^{3/2} D_0 \qquad [3\text{-}7]$$

where ε is the volume fraction of the conducting phase, and $\kappa_0$ and $D_0$ represent the conductivity and diffusivity of the pure electrolyte.

Removing Tortuosity

The Bruggeman correction shows that the conductivity of the composite decreases more rapidly than the volume fraction, which is, of course, an undesirable effect. Considering equations [3-4] and [3-7] together, it is clear that the effect of volume fraction on diffusive transport is similar to the effect on the conductivity.

The exact form of the porosity correction for ionic conductivity depends strongly on the structure of the composite material. It is desirable to investigate an electrode structure that has a more favorable conductivity correction for volume fraction than the Bruggeman correction. One simple such structure consists of lamellar particles arranged parallel to the direction of current flow, for which the porosity corrections are linear in volume fraction:

$$\kappa_{eff} = \varepsilon \kappa_0 \qquad [3\text{-}8]$$

$$D_{eff} = \varepsilon D_0 \qquad [3\text{-}9]$$

At an electrolyte volume fraction of 30%, the conductivity will be 30% of the liquid-phase conductivity in the case of the lamellar particles, but only 16% of the liquid-phase conductivity for the random porous structure with the Bruggeman correction. With such a lamellar microstructure, then, it should be possible to increase the volume fraction of active material without sacrificing ionic conductivity. Alternately, an electrode with identical particle loading should be able to deliver greater power, due to the improved ionic conductivity in the electrode.

Changing pore distribution

Another means of improving ion transport in the composite electrode is by adjusting the ionic conductivity to best respond to the current distribution in the electrode. When charge transfer current into the electrode particles is limiting, the current carried by the electrolyte phase in the electrode decreases with depth. This decreasing current with electrode depth indicates that the ionic conductivity of the electrolyte phase near the back of the electrode may not be critical, while a high ionic conductivity near the electrode surface is required to provide rapid ion transport towards the back of the electrode. Based on this need for varying ionic conductivity with electrode depth, it may be possible to improve the electrode rate capability without sacrificing utilization by grading the porosity of the electrode. A high volume fraction of electrolyte near the surface of the electrode would provide a good ionic conductivity in the region where the ion current is high to improve rate capability, while a higher fraction of active material in the depth of the electrode allows for the maintenance of a high energy density.

### Feasibility Study

A preliminary analysis was conducted to evaluate whether this proposed microstructure could provide improvements in battery performance. The effect of the graded microstructure on the ohmic loss in the electrolyte phase was approximated by using limiting cases for the ionic current distribution. Only ohmic loss in the liquid phase was considered in order to simplify this investigation. A reduction of ohmic loss in the battery would lead to a lower internal resistance and therefore lower polarization losses in the system. With a lower polarization, the cell should provide a higher specific energy when discharged to a given cutoff voltage.

Three different porosity gradients were considered and compared with a conventional electrode with a constant electrolyte volume fraction of 0.3. Each graded electrode had a spatially varying porosity while keeping total electrolyte volume fraction at 0.3. An electrode with a linear porosity gradient with an electrolyte fraction of 0.4 at the front of the electrode and an electrode fraction of 0.2 at the back of the electrolyte and electrodes with parabolic porosity gradients were investigated. Both a 'concave up' and a ‘concave down’ parabola were constructed with an average electrolyte fraction of 0.3 and an electrolyte fraction of 0.4 at the front of the electrode. In Figure 3.1, each of these porosity functions is shown. The functional forms of these respective pore gradients are: For a linear porosity gradient:

$$\varepsilon(d) = 0.4 - 0.2\left(\frac{d}{X}\right) \qquad [3\text{-}10]$$

For the ‘concave up’ parabolic porosity gradient:

$$\varepsilon(d) = 0.25 + 0.15\left(\frac{X-d}{X}\right)^2 \qquad [3\text{-}11]$$

For the ‘concave down’ parabolic porosity gradient:

$$\varepsilon(d) = 0.4 - 0.3\left(\frac{d}{X}\right)^2 \qquad [3\text{-}12]$$

In each case, ε(d) is the electrolyte volume fraction as a function of position *d*, and X is the total depth of the composite electrode.

First, a reaction front current distribution was assumed. In this limiting case, all of the electrochemical reaction would first take place at the electrode surface, until all of the oxide there had been completely intercalated with lithium. The reaction would then move into the depth of the electrode, intercalating the cathode material. In this situation, the ionic current would be equal to the cell current behind the reaction front, and zero ahead of the reaction front. This reaction distribution is to be expected in the case where liquid phase ion transport is limiting.

In this case, the ohmic loss in the liquid phase will change with time as the reaction moves from the front to the back of the electrode. The ohmic loss is determined by the total ionic resistance from the front of the electrode to the reaction location. If the reaction is taking place at a distance d from the front of the electrode, the ohmic loss, Δφ, attributed to liquid phase ion transport will be given by:

$$\Delta\varphi = \int_0^d \frac{i}{\varepsilon(x)^{3/2}\kappa_0}dx \qquad [3\text{-}13]$$

where x=0 is the front of the electrode, *i* is the cell current, $\kappa_0$ is the bulk ionic conductivity of the electrolyte, and ε is the volume fraction of the electrolyte phase as a function of position in the electrode.

In Figure 3.2, a figure of potential drop versus electrode depth is shown for this case. Each of the graded electrodes has a lower cumulative potential drop than the conventional electrode at the surface and throughout most of the electrode. However, the conventional electrode has a lower potential drop than all of the graded electrodes at the back end. Thus, the graded porosity electrodes would offer superior rate capability at the beginning of the discharge; after material near the surface is consumed, however, the conventional electrode would begin to exhibit superior performance as the material near the back of the electrode is consumed.

The second limiting case for the current distribution is one in which the ion current density decreases linearly across the electrode. This implies that the electrochemical reaction is evenly distributed throughout the electrode. In this case, the cumulative ohmic loss is calculated by multiplying the local ionic current by the local resistivity. The potential drop at a depth *d* is given by:

$$\Delta\varphi = \int_0^d \frac{i\left(1-\frac{x}{X}\right)}{\varepsilon(x)^{3/2}\kappa_0}dx \qquad [3\text{-}14]$$

where x is position, and X is the total electrode thickness.

Figure 3.3 shows the ohmic drop as a function of position in the electrode for this case. The graded porosity electrodes have a lower total potential drop than the uniform electrodes, by 8.5% for the 'concave up' parabolic porosity gradient, by 11.1% for the linear porosity gradient and by 13.7% for the 'concave down' porosity gradient. Near the electrode surface, where the ionic current is highest, the graded porosity electrodes have higher electrolyte volume fractions, and thus higher conductivity. At the rear of the electrode, the ion current is lower, and thus the ionic conductivity is not as critical. From this analysis, the graded porosity electrodes seemed to offer the potential for better rate capability. From these results, a more thorough modeling effort was clearly warranted.

## 3.2 Modeling

Review

To determine the impact of microstructural changes on real battery systems, the systems were modeled using computer simulations. Doyle, Fuller and Newman[21] have developed the most complete model for lithium batteries to date. Their model treats a full cell sandwich, including a porous cathode, an electrolyte separator, and either a lithium foil or a porous insertion anode. The model also incorporates concentrated solution theory, charge transfer processes and both electronic and ionic conduction[22] in the electrode. The model has been adapted to include a variety of secondary effects, such as temperature effects[23,24], different particle sizes[25], double-layer capacitance[26], as well as side reactions[27]. In addition, the group has provided guidelines for optimizing batteries for porosity and thickness to meet specific applications[28,29].

However, none of these previous electrochemical models considered the battery microstructure in detail. Nagarajan et al.[30] examined the effect of introducing a particle size distribution (PSD) in the electrode. The focus of the study was to assess the change in packing density that was attainable by using a PSD rather than single size particles. They found that by including a PSD, the capacity of cells was increased due to a higher packing fraction of active material in the electrode, resulting in a trade-off between solid-state and liquid phase diffusion limitations as the particle size and packing fraction increased.

## Description of the Starting Model

The model of Doyle, Fuller and Newman was adapted and modified. The source code and input files for this model were obtained at the Newman group web site at http://www.cchem.Berkeley.edu/~jsngrp/fortran.html

The original code is also included in this work as Appendix A. A detailed summary of this modeland the computational methods used to solve the set of partial differential equations that govern the battery system can be found in published literature [16,22].

The model uses porous electrode theory developed by Newman and Tobias[31] to simplify the complex cell geometry into a pseudo-one dimensional form. Equations for the lithium material balance [3-15], the potential in the liquid phase [3-16], charge transfer kinetics [3-17], the potential in the solid material [3-18], solid-state diffusion [3-19], and a conservation equation relating the charge transfer flux to the ion current in the liquid phase [3-20] are included[22]:

$$\varepsilon\frac{\partial c}{\partial t}=\frac{\partial}{\partial x}\left[D_{eff}\frac{\partial c}{\partial x}\right]+\left[1-t_{+}^{0}\right]aj_{n}-\frac{i_{ion}}{F}\frac{\partial t_{+}^{0}}{\partial x} \qquad [3\text{-}15]$$

$$\frac{\partial\Phi_{liq}}{\partial x}=-\frac{i_{ion}}{\kappa_{eff}}+\frac{RT}{F}\left[1+\frac{\partial\ln f_{A}}{\partial\ln c}\right]\left(1-t_{+}^{0}\right)\frac{\partial\ln c}{\partial x} \qquad [3\text{-}16]$$

$$j_{n}=\frac{i_{o}}{F}\left[\exp\left[\frac{\alpha_{a1}F\left(\Phi_{sol}-\Phi_{liq}\right)}{RT}\right]-\exp\left[-\frac{\alpha_{c1}F\left(\Phi_{sol}-\Phi_{liq}\right)}{RT}\right]\right] \qquad [3\text{-}17]$$

$$\frac{\partial\Phi_{sol}}{\partial x}=\frac{I_{cell}-i_{ion}}{\sigma_{eff}} \qquad [3\text{-}18]$$

$$\frac{\partial c_{sol}}{\partial t}=D_{sol}\left[\frac{\partial^{2}c_{sol}}{\partial r^{2}}+\frac{2}{r}\frac{\partial c_{sol}}{\partial r}\right] \qquad [3\text{-}19]$$

$$aj_{n}=\frac{1}{F}\frac{\partial i_{ion}}{\partial x} \qquad [3\text{-}20]$$

where $f_A$ is the thermodynamic factor used for non-ideal solutions, $c_{sol}$ and $\Phi_{sol}$ are the concentration and potential in the solid phase, $i_{ion}$ is the liquid phase ion current, $I_{cell}$ is the total cell current, and F is the Faraday constant. The tortuosity corrections for D and κ are those in equations [3-6] and [3-7]. These equations are linearized and solved using finite differences. The Crank-Nicholson scheme is used for time derivatives. For terms involving the second derivative, a central difference form is used, while either forward or backward differences are used for terms involving first derivatives in space. The equations are cast into matrix form, and solved by the method detailed in the literature[14].

The model calculates the specific energy and power of the cell, using a cell mass M that includes the mass of the anode, cathode, separator and both current collectors. The mass of any external packaging is not included. The specific energy *E* of the cell is given by[21]

$$E = \frac{1}{M}\int_0^t IVdt \qquad [3\text{-}21]$$

where I is the cell current, V is the cell voltage. The specific power *P* of the cell is given by[21]

$$P = \frac{E}{t} \qquad [3\text{-}22]$$

Of particular interest is the treatment of the geometric parameters in this model, and their effect on the simulations. The intercalation oxide particle size is used, along with the oxide volume fraction to calculate the surface area *a* available for the electrochemical reaction. The particle size also enters into the solid-state diffusion calculations. The geometry of the oxide particles does not enter into any direct calculations of the liquid-phase ion transport.

The other geometric parameter, the phase fraction is used to calculate the effective liquid phase diffusion coefficient and conductivity. The volume fraction of solid phase is also used to calculate the particle surface area and the solid phase conductivity. The solid-state diffusion coefficient is not modified for tortuosity, because solid-state diffusion occurs only on a local scale, into and out of the oxide particles.

Other than these microstructural parameters, the model also requires inputs for the dimensions of each battery layer, the density of each of the material components, temperature, electronic conductivity of each electrode material, kinetic constants for the electrochemical reactions, coulombic capacity, and the electrochemical testing conditions. The input parameters that were kept constant for all of the simulations are listed in Table 3.1. A complete sample input file can be found in Appendix B.

Table 3.1 Input parameters consistent through each simulation

| **Input parameter** | **Value** |
|---|---|
| Anode thickness | 100 μm |
| Separator thickness | 52 μm |
| Cathode thickness | 200 μm |
| Positive current collector thickness | 25 μm |
| Negative current collector thickness | 25 μm |
| Temperature | 298.13 K |
| Initial salt concentration (in electrolyte) | 2.0 [M] |
| Diffusion coefficient in anode solid (MCMB Carbon) | 3.9e-10 $cm^2/s$ |
| Diffusion coefficient in cathode solid ($Li_xMn_2O_4$) | 1e-9 $cm^2/s$ |
| Radius of anode particles | 12.5 μm |
| Radius of cathode particles | 8.5 μm |
| Conductivity of anode solid | 100 S/m |
| Conductivity of cathode solid | 3.8 S/m |
| Rate constant for charge transfer reaction at anode | 5e-9 |
| Rate constant for charge transfer reaction at cathode | 5e-9 |
| Capacity of anode Li storage material | 372 mAh/g |
| Capacity of cathode Li storage material | 148 mAh/g |

| **Input Parameter** | **Value** |
|---|---|
| Volume fraction electrolyte in anode | .357 |
| Volume fraction polymer in anode | .146 |
| Volume fraction filler in anode | .026 |
| Volume fraction electrolyte in separator | .724 |
| Volume fraction polymer in separator | .276 |
| Average volume fraction electrolyte in cathode | .300 |
| Volume fraction polymer in cathode | .000 |
| Volume fraction filler in cathode | .073 |
| Density of electrolyte | 1.324 g/cm$^3$ |
| Density of anode insertion material | 1.900 g/cm$^3$ |
| Density of cathode insertion material | 4.140 g/cm$^3$ |
| Density of inert filler (conductive support) | 1.800 g/cm$^3$ |
| Density of polymer | 1.780 g/cm$^3$ |
| Density of separator material | 2.000 g/cm$^3$ |
| Density of negative current collector | 8.930 g/cm$^3$ |
| Density of positive current collector | 2.700 g/cm$^3$ |
| Initial cathode material stoichiometry (x in $Li_xMn_2O_4$) | 0.1705 |
| Initial anode material stoichiometry (y in $Li_yC_6$) | 0.5635 |

During the simulations, the volume fraction of electrolyte in the cathode as a function of depth was varied. However, in each case, the average volume fraction of electrolyte in the cathode was kept constant at 0.3. In addition to the electrolyte volume fraction, the discharging and charging current were varied between tests. A schematic representation of the simulated cell with relevant cell dimensions is depicted in Figure 3.4.

It should be noted that although this battery model is a powerful tool for battery design and simulation, it has some limitations in accurately depicting microstructural complexity. Because the model casts the battery system into one dimension, it may falter when there is significant transport parallel to the current collectors, as would be expected for instance for an electrode configuration composed of parallel platelet particles, arranged parallel to the superficial current flow.

### Modifications to the model of Doyle, Fuller and Newman

Modifying the code to simulate particles with a linear porosity correction for tortuosity required changing the Bruggeman exponent from 1.5 to 1.0 in accordance with equations [3-8] and [3-9]. This change affected the calculation of the effective diffusion coefficient and ionic conductivity, but did not fundamentally change the complexity of the model or the governing equations.

Simulating electrodes with a graded porosity required several changes to the code. First, the volume fraction of each phase was stored as an array in order to represent spatial variance of porosity. Because the value of porosity affects the specific surface area and electronic conductivity, which had been invariant in space in the original model, these parameters also were recast into arrays. The volume fraction also enters the governing equations in differential form in the material balance equation [3-15].

The left hand side of the equation need not be modified, since the time derivative of the volume fraction is zero. However, the electrolyte volume fraction enters the left hand side of the equation through the effective diffusion coefficient, $D_{eff} = \varepsilon^{3/2} D_0$. Equation [3-15] then becomes:

$$\varepsilon \frac{\partial c}{\partial t} = \frac{\partial}{\partial x}\left(\varepsilon^{3/2} D \frac{\partial c}{\partial x}\right) + \left(1 - t_+^0\right) a j_n - \frac{i}{F}\frac{\partial t_+^0}{\partial x} \qquad [3\text{-}23]$$

Using the product rule, the equation can be further expanded

$$\varepsilon \frac{\partial c}{\partial t} = \frac{3}{2}\varepsilon^{1/2}\frac{d\varepsilon}{dx} D \frac{\partial c}{\partial x} + \varepsilon^{3/2}\frac{\partial D}{\partial x}\frac{\partial c}{\partial x} + \varepsilon^{3/2} D \frac{\partial^2 c}{\partial x^2} + \left(1 - t_+^0\right) a j_n - \frac{i}{F}\frac{\partial t_+^0}{\partial x} \qquad [3\text{-}24]$$

For an electrode with a constant porosity, the first term on the right hand side is zero, because dε/dx is zero. For a graded electrode, this term is nonzero, and was therefore added to the solution scheme for the differential equations. Once the code was successfully modified, it was compiled and run on Sun Ultra5 and SparcStation5 Unix workstations. Depending on the conditions for the simulation, run time for the simulation of a single cell charge or discharge was between 1 and 30 minutes.

The code was modified to allow the simulation of arbitrary linear gradients in porosity by changing the input file. The electrolyte volume fraction at the front and back of the electrode are new input parameters. Quadratic porosity gradients were hard-coded for simplicity. The Fortran code as modified for the simulation of batteries with linear porosity gradients is given in Appendix C.

## 3.3 Results

In order to verify the accuracy of the code modifications, the modified code was first run for an electrode of constant porosity, and the results compared with those obtained from the original code for identical simulation parameters. The results before and after modification of the code matched precisely for a variety of simulation conditions. The simulations were run using materials properties for $LiMn_2O_4$ spinel cathode material, the EC/DEC/$LiPF_6$ electrolyte system, and either MCMB carbon or lithium anodes. The MCMB carbon anode was used for tests involving graded electrodes. The lithium metal anode was used for the simulations in which the tortuosity had been removed from the electrode. These systems were chosen because the most complete materials data were available.

For all discharge tests, a spinel cathode was assumed with an initial lithium content of $Li_{.1705}Mn_2O_4$. This material was discharged galvanostatically to a cutoff potential of 3.5 V in all simulations. A variety of current densities and graded electrodes were used. The total electrolyte volume fraction in the cathode was kept constant at 0.3 for all electrodes tested in the model. The layer thicknesses assumed in these simulations were 200 μm for the cathode, 52 μm for the separator, and 100 μm for the anode. Comparisons were made between cells with identical separator and anode configurations in order to isolate the effects of modifying the cathode microstructure.

In the discussion of modeling results, it was frequently necessary to refer to the region of the cathode near the cathode-separator interface, as well as to the region of the cathode near the current collector. The former will be referred to as the front or surface of the electrode, while the latter will be called the back or bottom of the electrode.

Electrodes with linear porosity gradients

Cathodes with a variety of linear porosity gradients were simulated against an MCMB Carbon anode with constant porosity. Tests were performed on cathodes with high electrolyte volume fractions near the electrode surface, with constant electrolyte fraction, and also on cathodes with a low electrolyte fraction near the front of the electrode. Cells were simulated under conditions of a constant current discharge with a cutoff potential of 3.5V. The cathodes were simulated such that the average electrolyte volume fraction in the electrode remained constant at 0.3. In order to easily refer to electrodes with a particular linear porosity gradient, a graded electrode will be referred to by the electrolyte volume fraction at the front and back of the electrode. For example, a .4/.2 electrode would be an electrode with an electrolyte volume fraction of 0.4 at the front of the electrode and an electrode fraction of 0.2 at the back of the electrode and a linear gradient in electrolyte fraction from the front to the back of the electrode. An electrode with a constant electrolyte volume fraction will be referred to as conventional or homogeneous, and will serve as a benchmark in measuring performance for the graded electrodes.

Figure 3.5 shows the specific energy vs. discharge current for electrodes with linear porosity gradients and an average electrolyte volume fraction of 0.3. At low rates, the performance of the electrodes is virtually identical; here, the electrodes are limited by the intrinsic capacity of the oxide. As the discharge rate is increased, liquid phase ion transport in the electrode becomes limiting, and the graded porosity electrode improves cell performance. At 10 A/m$^2$, the .4/.2 electrode offers 30% greater specific energy than the conventional electrode. This increase in specific energy is achieved without sacrificing specific power. The specific energy decreases with increasing current for all electrodes. However, the electrodes with graded porosity retain a higher specific energy as the current is increased. At 20 A/m$^2$, the .4/.2 electrode has a specific energy 60% higher than an electrode with constant porosity. At this current density, a .5/.1 electrode twice the specific energy of a homogeneous electrode. In order to verify that the benefits of grading the electrode occur because of the need for higher liquid phase ion transport near the electrode surface, 0.2/0.4 electrodes were also simulated. These electrodes are expected to have lower specific energies than a conventional electrode, because the low electrolyte fraction near the front of the electrode would decrease the ionic conductivity in this region. As expected, the simulations showed that the 0.2/0.4 electrode had a lower specific energy than the conventional electrode for all discharge current densities.

A Ragone plot of specific power vs. specific energy for the same electrodes is shown in Figure 3.6. Here, the benefits of grading the electrode are made more apparent as the 'knee' of the Ragone plot is moved outward. The graded porosity electrode is able to supply more energy at a given power than the conventional electrode. It should be noted that the 'knee' of the plot represents the ideal operational range for the cell. At this location, the optimum tradeoff between specific energy and specific power occurs. Therefore the figure indicates that the improvement in capacity/rate tradeoff due to electrode grading occurs in a useful performance range.

Figure 3.7 shows the specific energy as a function of linear electrode grading for several different discharge currents. As the discharge current increases, the optimum electrode grading shifts from a slight porosity gradient to more severe porosity gradients at very high current densities. At 5 A/m$^2$, the highest specific energy is obtained with a .375/.225 electrode. At 10 A/m$^2$, the .45/.15 electrode has the highest specific energy. The .599/.001 electrode has the highest specific energy at 20 A/m$^2$. This shift in the optimum grading can be attributed to the decreasing electrode utilization with increasing current. As the depth of the electrode is less utilized at high currents, the steeply graded electrode, with its superior ion transport near the electrode surface, becomes more effective than more gradually graded electrodes. Conversely, as other electrodes are more fully utilized at lower currents, the poor ion transport properties at the back of the highly graded electrode prevent full utilization at low and moderate discharge rates.

Electrodes with non-linear porosity gradients

Figure 3.8 shows a comparison of specific energy vs. discharge current for conventional electrode of electrolyte volume fraction 0.3 with electrodes of linear and quadratic porosity gradients. Cell simulations were run under constant current with a cutoff potential of 3.5 V. The anode was a composite MCMB carbon electrode. Equations [3-10] through [3-12] describe the electrolyte volume fraction as a function of depth for each electrode. Each of the graded electrodes has an electrolyte volume fraction of 0.4 at the electrode surface, and an average electrolyte volume fraction of 0.3. Both a 'concave up' and a 'concave down' quadratic gradient were simulated. All of the graded electrodes have a higher specific energy than a homogeneous electrode in the intermediate discharge rate regime. At very low discharge rates, the cell is limited by the electrode capacity, and there is little performance difference between any of the electrodes. At very high discharge rates, the performance of the graded electrodes also converges. This can be attributed to the fact that only the oxide near the electrode/separator interface is utilized at very high discharge rates. As all of the graded electrodes have identical 0.4 electrolyte volume fractions at the electrode surface, it is expected that their performance should be similar if only the electrode surface is utilized. In addition, although the constant porosity electrode has 50% lower specific energy than the graded electrodes at the highest discharge current of 40A/m$^2$, the absolute difference is small because none of the electrodes is well utilized. The graded electrodes have specific energies within 5% of one another at this highest value, (between 3.88 and 4.08 Wh/kg), whereas the constant porosity electrode yields 2.15 Wh/kg.

The most important distinctions in terms of performance between the graded electrodes occurs in the intermediate discharge rate regime. This type of behavior is not unexpected, because the electrodes have identical average and surface compositions. Thus, we expect that if all of the electrode is utilized (at low rates), or if only the surface is utilized (at high rates), the performance of the electrodes should be very similar. It is in the intermediate range, where the electrode is partially utilized, that performance differences occur. The electrode with a 'concave down' parabolic gradient should be expected to perform best, as it has the highest volume fraction of electrolyte near the electrode surface. Indeed, this electrode has a 10% higher specific energy than the linear .4/.2 electrode at a discharge rate of 15 A/m$^2$.

### Simulations of cell charging

Simulations were also performed to test the performance of graded electrodes during battery charging. During charging, the battery is connected to some power source, presumed to be large, and the lithium is deintercalated from the cathode, and redeposited in the anode of the cell. Because the battery is charged by an external power source, the energy efficiency of the charging process is not as critical as the charge rate and the amount of lithium that is deintercalated from the cathode. Cell charging was simulated using data for a spinel cathode with an initial lithium stoichiometric parameter of 0.75 and an MCMB composite anode. The cell dimensions were identical to those used in the discharge simulations. The cells were charged under constant current to a 4.3 V cutoff potential. In Figure 3.9, the stoichiometric utilization of the electrode is shown as a function of charging current for both a graded and a homogeneous electrode.

The charging curves validate the results of the discharge simulations. It is clear that the graded electrode again performs better than the homogeneous electrode, with a higher utilization at moderate and high discharge rates. During charging, the ionic current will be distributed as it is during discharge, but in the opposite direction. Therefore, it is again imperative to have high ionic conductivity near the electrode surface where the ionic current is the greatest.

### Simulation of electrodes with tortuosity removed

Discharge simulations were also performed on electrodes in which the tortuosity was eliminated. With this cell geometry, the Bruggeman correction is substituted with a linear correction for volume fraction, as shown in equations [3-8] and [3-9]. Attempts were made to simulate solid particles with plate-like geometry. However, the code did not function properly with this modification. Instead, the existing code was used with spherical particles. While this situation is not physical, it does provide some insight into the operation of electrodes without tortuosity.

In Figure 3.10, the specific energy is shown as a function of discharge current for a porous electrode and electrodes without tortuosity. In these simulations, a Li foil anode was used instead of an MCMB Carbon anode because altering the Bruggeman correction in only one of the porous electrodes was not possible. By using a Li foil electrode, the effect of modifying the cathode was isolated. At discharge currents up to 50 A/m$^2$, the electrode without tortuosity was utilized at over 90% efficiency. At this current density, the conventional porous electrode has a specific energy 85% lower than that of the electrode without tortuosity.

Because platelet shaped particles of a given thickness have a lower surface area and a larger diffusion distance than spherical particles with a radius of the same dimensions, simulations were also performed on a tortuosity-free electrode with triple the oxide particle size. This particle size was chosen because the surface area for these particles was the same as for platelet-shaped particles of the original particle size. In addition, the diffusion distance for these particles was larger than for platelet-shaped particles of the original particle size. However, despite reducing the particle surface area and increasing the diffusion distance, the tortuosity-free electrode with a large particle size retained much of the improved rate-capability when compared with a porous electrode. At 50 A/m$^2$, this electrode with large particles had a specific energy more than six times higher than that of the porous electrode.

The data comparing the porous electrode to a tortuosity-free electrode is also displayed as a Ragone plot in Figure 3.11. The chart shows that by removing tortuosity, liquid-phase ion transport is improved such that the discharge capacity of the intercalation oxide limits the cell up to current densities of 50 A/m$^2$.

## 3.4 Conclusions

Ion transport in the lithium battery was analyzed in order to develop microstructural improvements to battery systems. Solution-phase ion transport in the composite electrode was determined to be the rate-limiting step at high rates. Two methods of ameliorating this transport step were proposed and tested using the battery model developed by Doyle, Fuller and Newman.

Cell modeling on both graded porosity electrodes as well as electrodes with removed tortuosity showed that specific energy at a given discharge rate could be increased by refining the electrode microstructure. Utilizing graded porosity electrodes offered an increase in specific energy of 30% over a homogeneous electrode at a discharge rate of 10A/m$^2$. An electrode without tortuosity offered a specific energy more than six times greater than that of a homogeneous porous electrode at a discharge rate of 50 A/m$^2$.

Clearly, these modeling results are promising and merit experimental confirmation. Though the intrinsic capacity of the lithium storage material provides an absolute limit on the energy density of the lithium ion battery, these microstructural improvements allow the use of significantly higher discharge rates while obtaining the same energy densities. The improved rate capability of microstructurally enhanced electrodes makes them suitable for high-power battery system. These simulations demonstrate that battery performance can be significantly improved through the optimization of electrode microstructure.

## 3.5 Figures

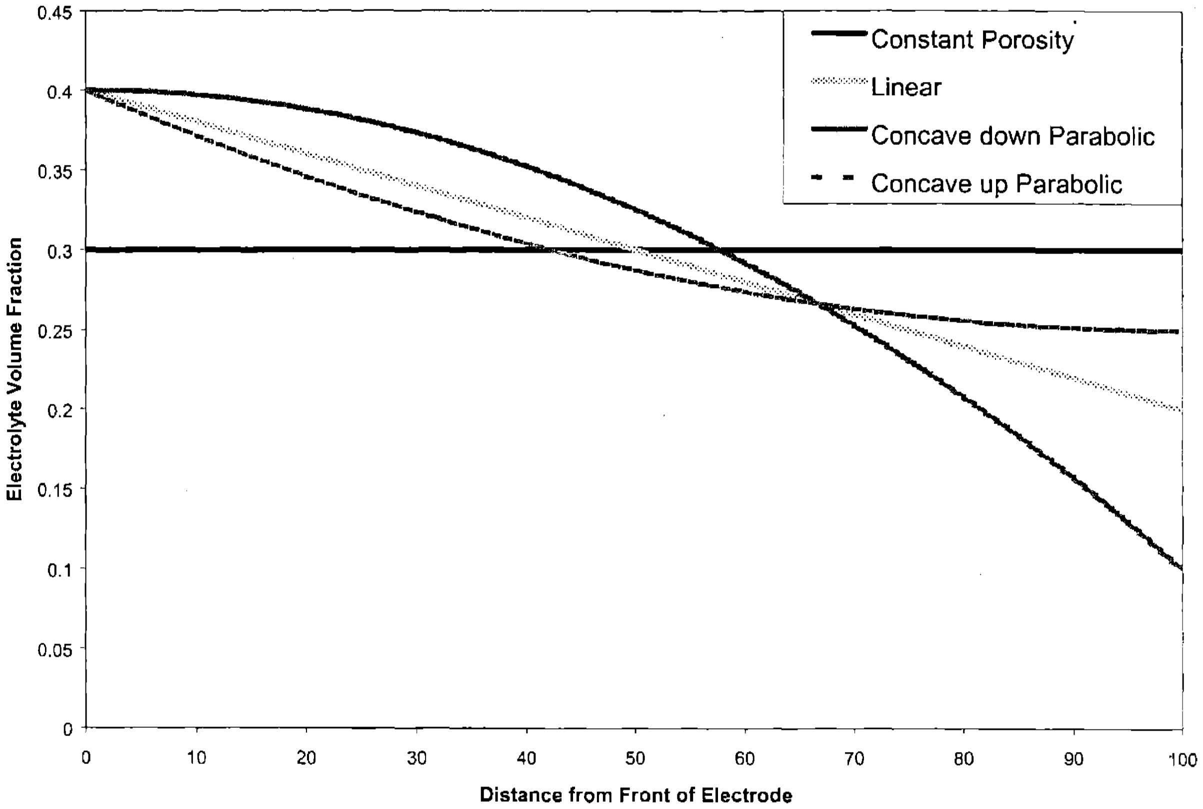

Figure 3.1 Electrolyte volume fraction as a function of distance from the front of the electrode for a selected electrodes with graded porosities. The average electrolyte fraction for each electrode was 0.3. Each of the graded electrodes had an electrolyte fraction of 0.4 at the front of the electrode.

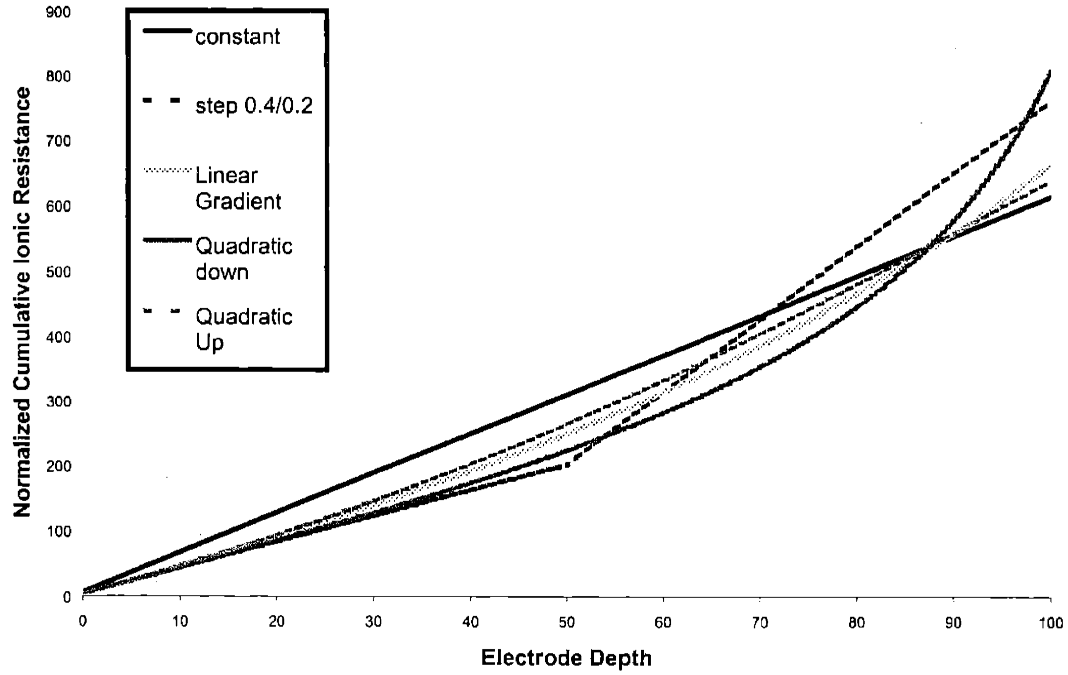

Figure 3.2 Normalized cumulative ionic resistance as a function of depth in the electrode for a variety of graded electrodes. Each graded electrode offers lower ionic resistance near the electrode surface; however, the homogeneous has the lowest total resistance at the back of the electrode.

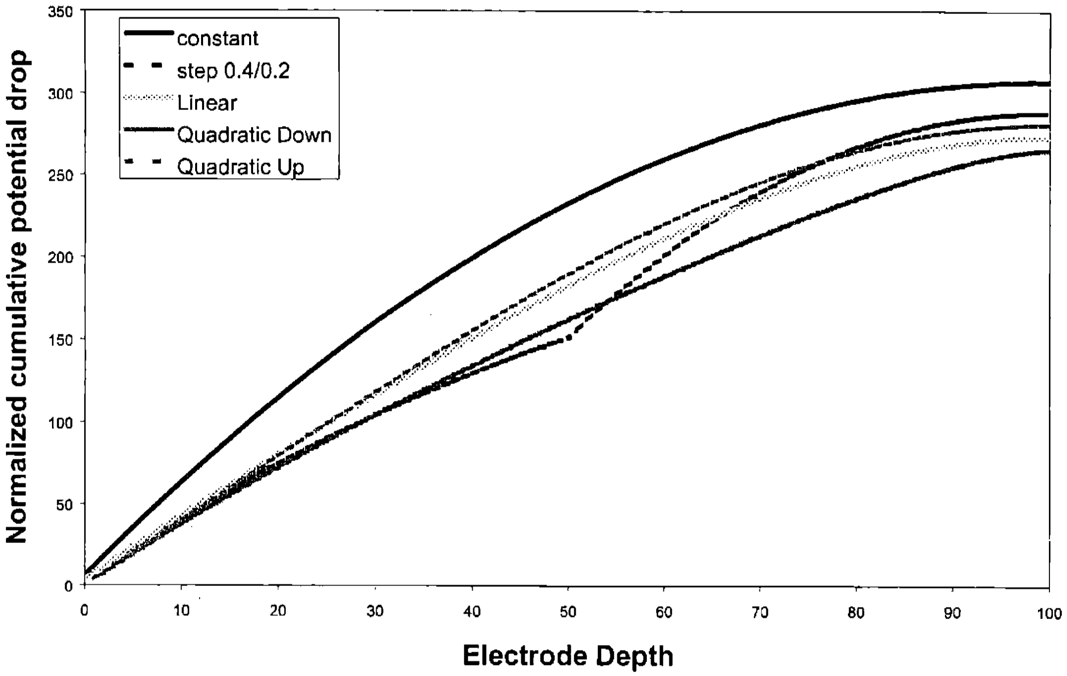

Figure 3.3 Normalized cumulative potential drop versus depth in the electrode for a variety of graded electrodes, assuming an even spatial distribution for the electrochemical reaction in the cell. All of the graded electrodes offer performance benefits (lower potential drop) when compared with the homogeneous electrode, even at the back of the electrode. This benefit accrues because the graded electrodes have higher ionic conductivity near the electrode surface, where the ionic current is greatest.

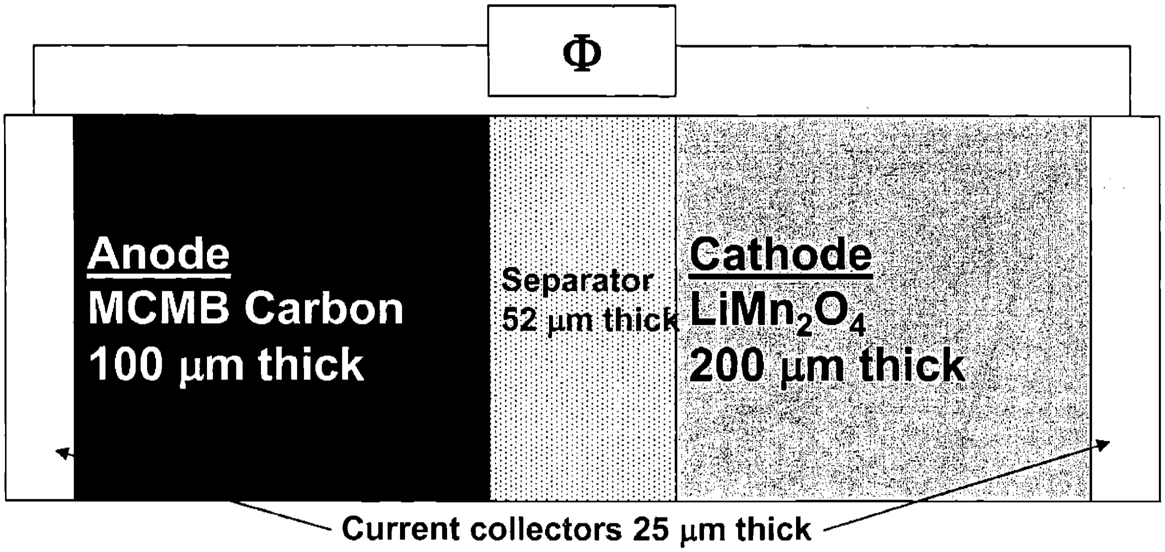

Figure 3.4 A schematic representation of the simulated cells. The dimensions of each cell component are noted.

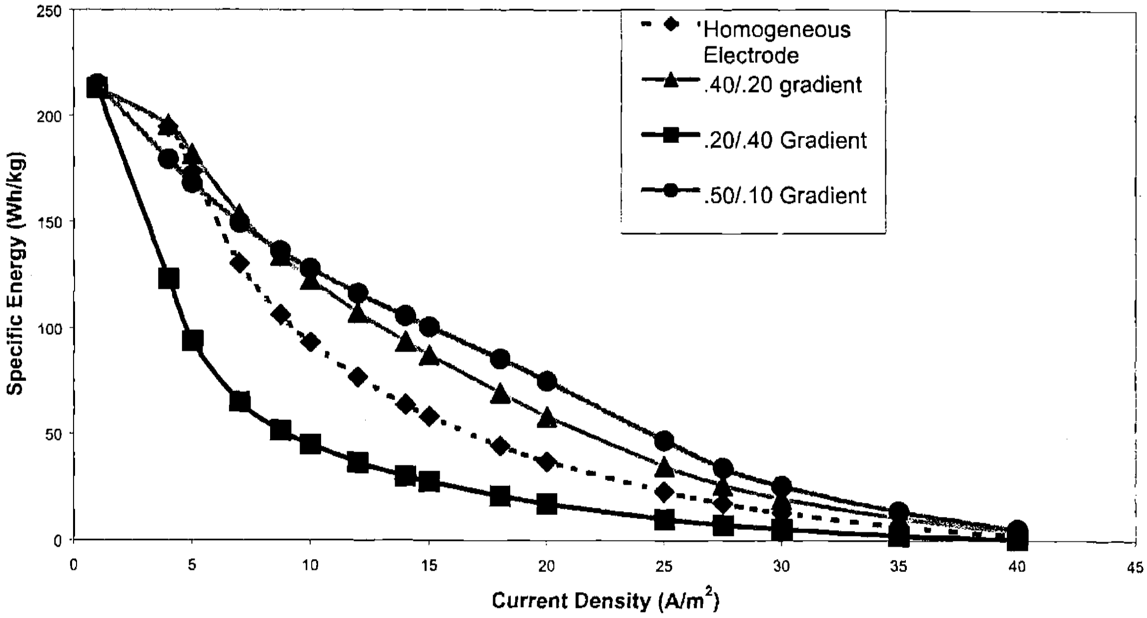

Figure 3.5 Specific Energy vs. Discharge Current for electrodes with a variety of linear porosity gradients. The electrode with the greatest volume fraction electrolyte near the electrode surface performed best at high discharge rates. The electrode with a modest porosity gradient allowed full utilization at low discharge rates, but also provided performance benefits over the homogeneous electrode at high discharge rates.

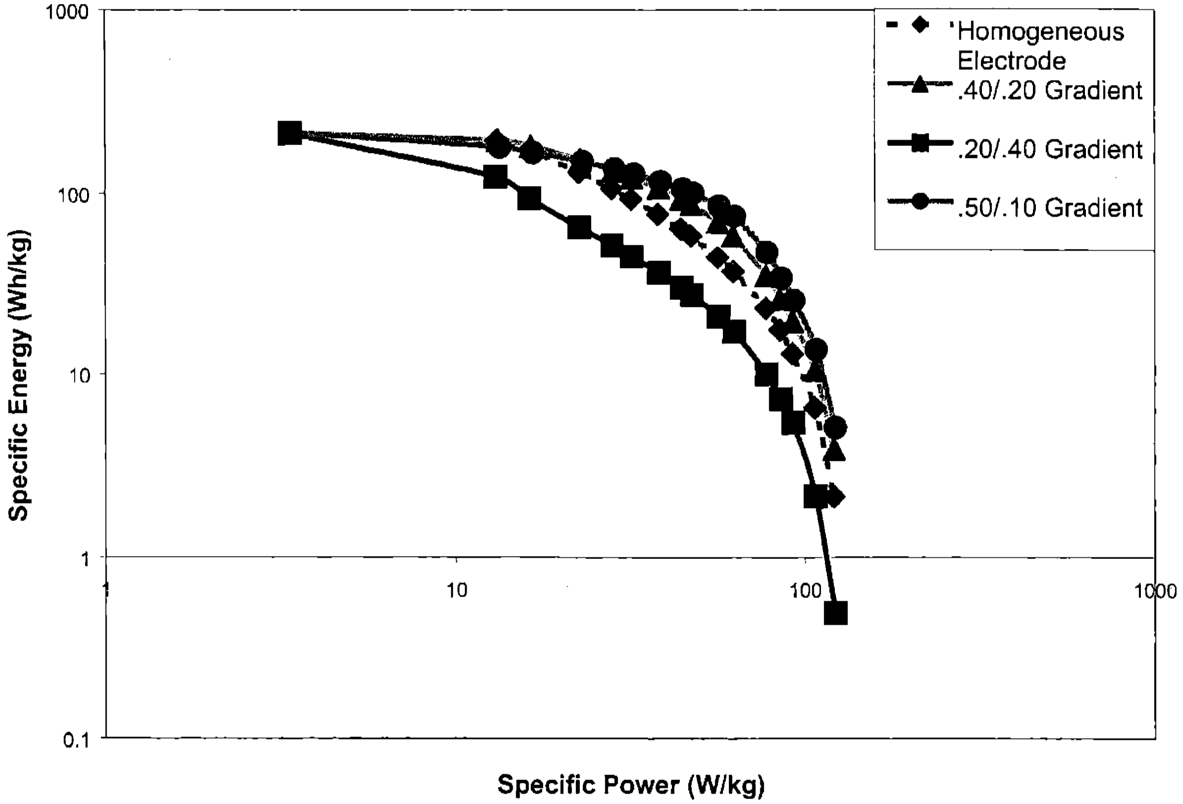

Figure 3.6 Ragone plots for electrodes with a variety of linear porosity gradients. The electrodes with higher electrolyte fractions near the electrode surface show better performance at high discharge rates, as shown by the higher specific energy available under high-power discharges.

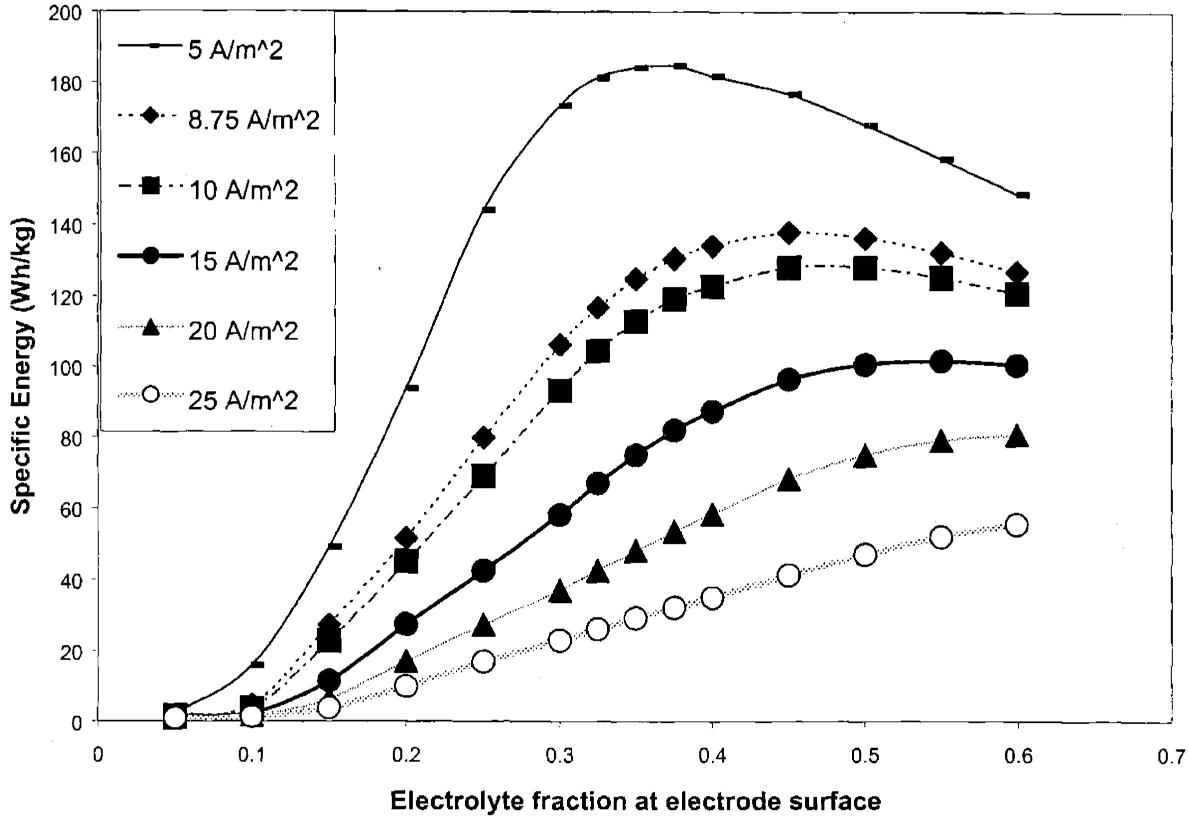

Figure 3.7 Specific Energy as a function of the porosity gradient for a variety of discharge current densities. The optimum electrode has from a modest gradient at low discharge rates but more severe gradients at high discharge rates. This result is consistent with the interpretation that at high rates, only the surface of the electrode is utilized.

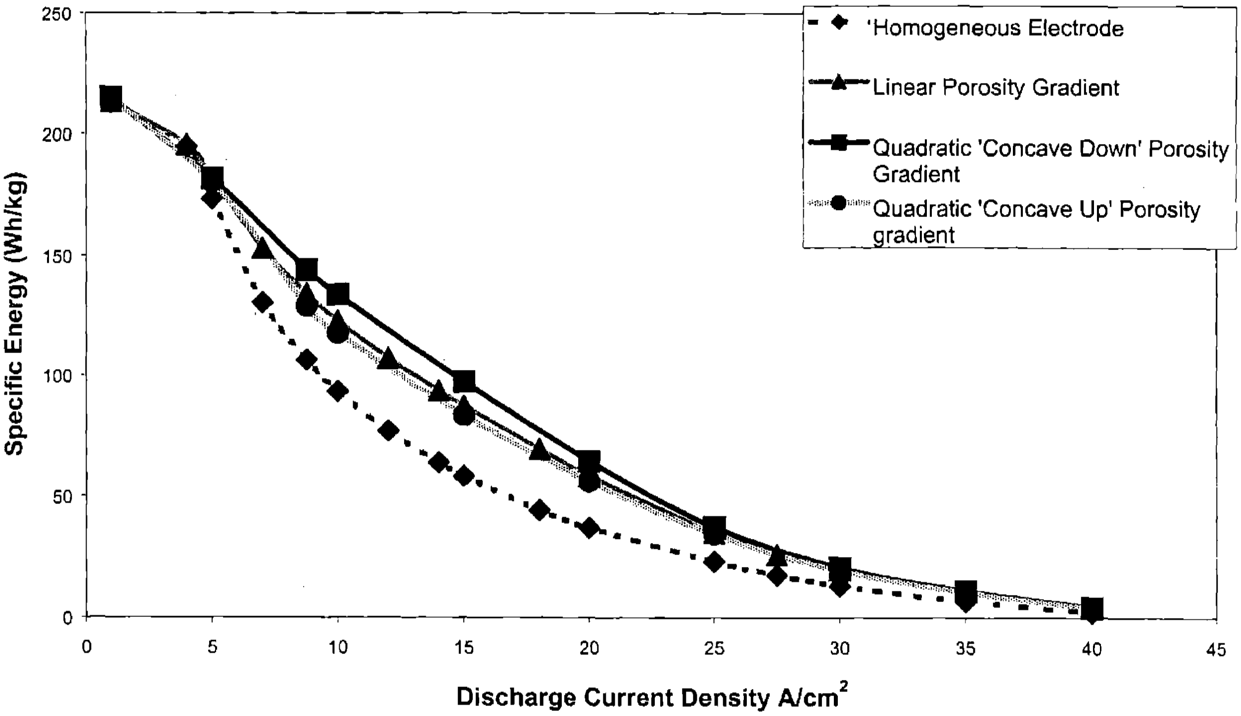

Figure 3.8 Specific Energy vs. Discharge Current Density for Electrodes with a Non-linear Porosity gradient, compared to a homogeneous and a linearly graded electrode. Each of the graded electrodes had an average electrolyte volume fraction of 0.3 and an electrolyte fraction of 0.4 at the electrode surface. At low rates, the performance of the electrodes is similar, because they are limited by the capacity of the intercalation material. At high rates, the graded electrodes also show very similar behavior because they have similar porosities near the electrode surface. In the intermediate current range, the electrode with a 'concave down' porosity gradient performs best, because near the surface, the electrode volume fraction decreases more slowly than for the other graded electrodes.

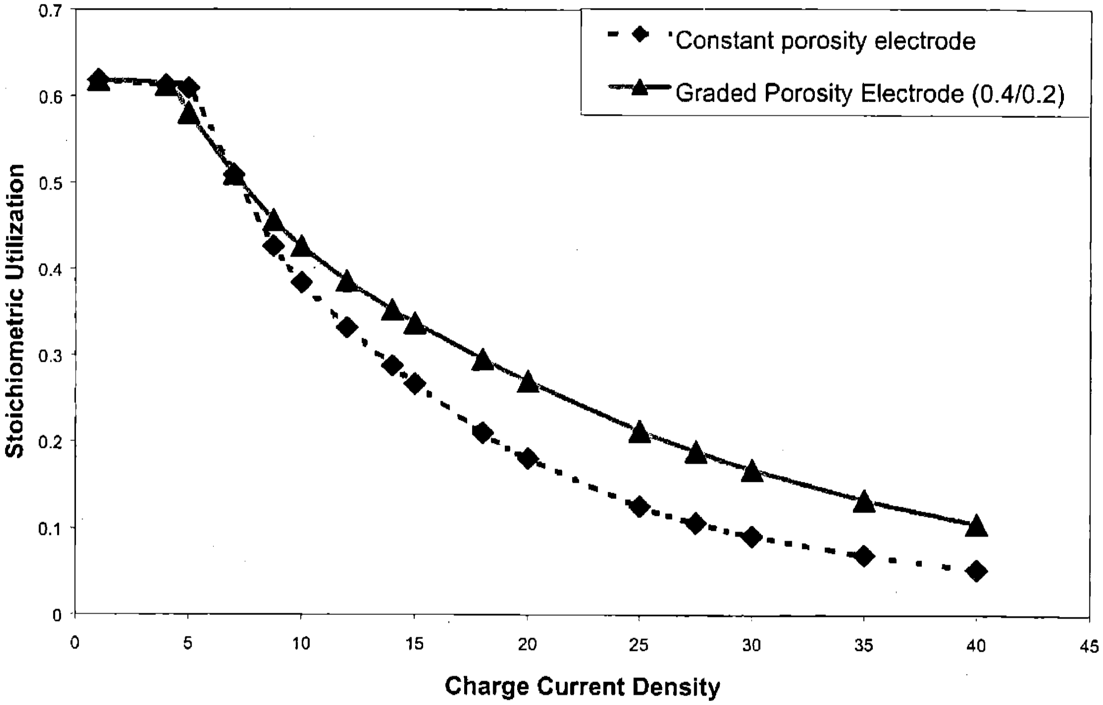

Figure 3.9 Stoichiometric Utilization as a function of charging current for an electrode with a linear porosity gradient compared with a homogeneous electrode. Grading the electrode increases specific capacity for both charging and discharging at moderate and high rates.

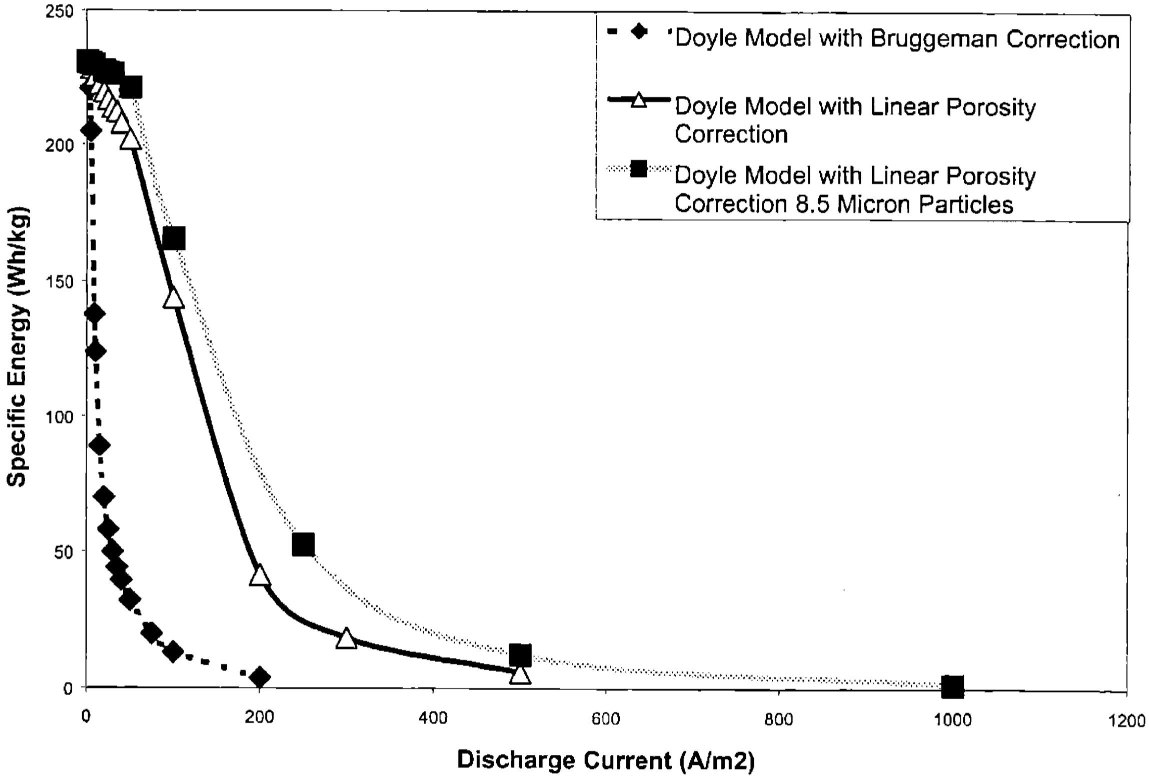

Figure 3.10 A tortuosity-free electrode is compared to a homogeneous porous electrode with the Bruggeman correction for tortuosity. Use of the linear rather than Bruggeman correction provides enhanced ion transport at moderate and high discharge rates.

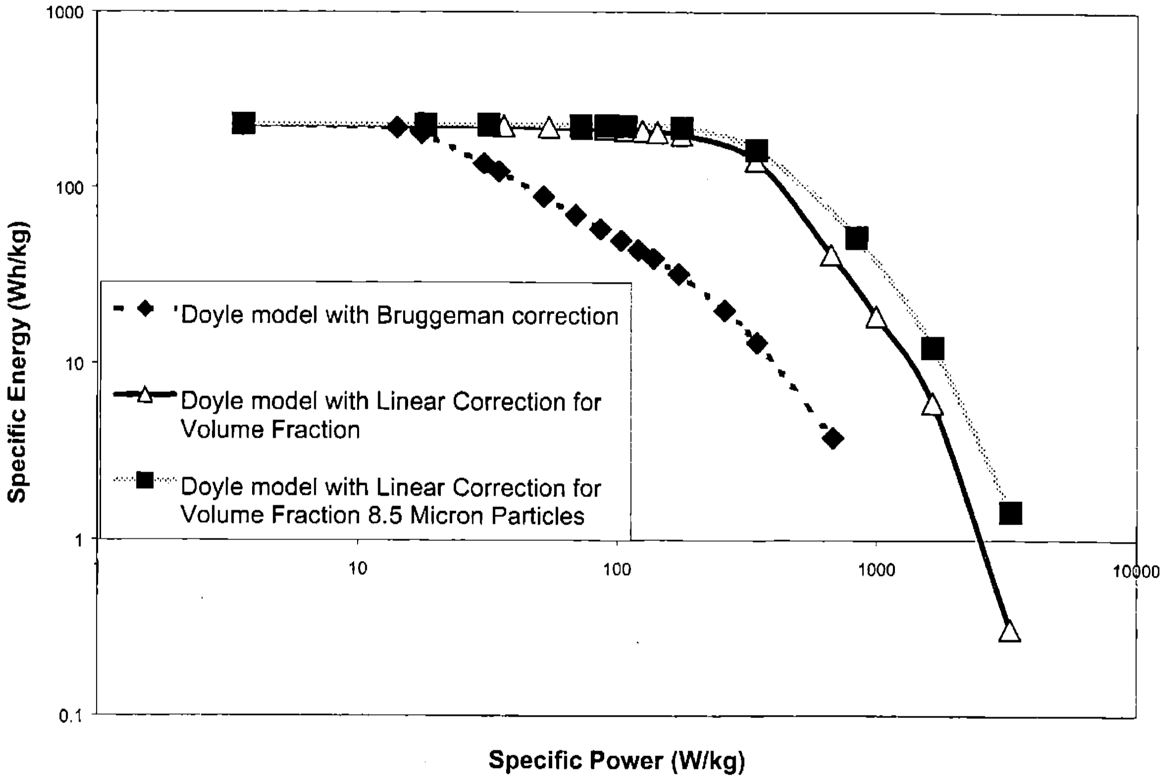

Figure 3.11 Ragone plot comparing a homogeneous electrode with an electrode with parallel plate particle configuration. The Ragone plot shows that the electrode with linear porosity correction has a higher rate capability than the porous electrode.

# 4. Conclusions

Transport of charged species in lithium ion batteries was investigated from a microstructural approach. The rate-limiting step for transport of each species was identified, and microstructural methods to address each limiting step were proposed.

Electron transport in the composite electrode was analyzed using percolation theory, and was compared to other conductor-insulator composite systems. An *in situ* filter pressing apparatus was designed and constructed in order to determine the percolation threshold for electron transport in battery electrode systems and to gain a qualitative understanding of the inter-particle interactions in these composites. The percolation threshold was determined to be between 10 and 13 volume percent solid for liquid electrolyte systems. It was found that systems with dissolved polymers had percolation thresholds at much higher solid loadings of 25 volume percent. This effect is attributed to polymer adsorption on solid conductor particles.

Ion transport was analyzed by considering each step in ion migration from the anode to the intercalation into the cathode oxide. The rate-limiting step was identified as liquid phase ion transport in the composite electrode. Microstructural solutions to alleviate this bottleneck in lithium transport were proposed and evaluated by modifying the simulation model developed by Doyle, Fuller and Newman. Simulation results showed that graded electrodes showed improved rate capability over homogeneous porous electrodes. This improvement is due to the higher ionic conductivity of the graded electrode near the electrode surface, where the ionic current is greatest. Performance improvements of 30% in terms of specific energy were achieved for .4/.2 graded electrodes at discharge rates of 10 A/m$^2$.

Electrodes with no tortuosity showed higher rate-capabilities than conventional porous electrodes. Such electrodes could be discharged at rates up to five times greater than conventional porous electrodes and without sacrificing capacity. At a discharge rate of 50 A/m$^2$, the tortuosity-free electrode had a specific energy more than six times as high as the porous electrode. Shortening the ion migration distance, improves the rate capability in these cells.

These microstructural refinements demonstrably improved the rate-capability of lithium battery electrodes. Experimental confirmation of these modeling results is justified by the increase in rate-capability seen in the simulations.

# 5. Appendices

## Appendix A – Doyle's code

```
c**********************************************************************
**
c
c   dual.f  (version 3.0)  January 10, 2000
c   Dual lithium ion insertion cell
c   Includes Bellcore physical properties
c Copyright Marc Doyle and John Newman 1998.
c You may make a copy of this program which you may
c personally and freely use in its unaltered form.
c You may distribute this program subject to the
c conditions that it be made freely available and
c that any duplication of this program must be
c essentially unaltered and must include this notice.
c
c We make no warranties, express or implied, that
c this program is free of errors or that it will
c meet the requirements of your application.  The
c author and publisher disclaim all liablility for
c direct or consequential damages resulting from
c use of this program.
c
c   Revised June, 1998, to include double-layer capacitance in
c   each electrode and to correct a factor of two in Ohm's law.
c
c Note: For lflag=0, the model works only for initially zero current.
c
c   Revised Feb. 12, 1999:
c   - if n1 = 0, then code treats the negative electrode as metal foil.
c   - subroutine cellpot does not calculate utilization of foil electrode
c   - Changed read and print statements.
c     To run, simply type "webdual", then enter
c     input and output file names when prompted.
c   - double layer capacitance is not currently calculated at
c     a foil electrode
c*******************************************************************
      implicit real*8(a-h,o-z)
      character *30 filin, filout
      parameter(maxt=900)
      common /n/ nx,nt,n1,n2,nj,n3,tmmax
      common /calc/ ai(maxt),ai2(maxt),u(223,maxt),ts(maxt),
     1h,h1,h2,h3,hcn,hcp,rr,rrmax
```

```
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     1ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1
      common/power/ ed,Vold,ranode,rcathde
      common/ssblock/ xp0(6),xx0(6,221),term(221),fj(221)
      common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
      common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,
     1sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
      common/tprop/df(221),cd(221),tm(221),
     1ddf(221),dcd(221),dtm(221),dfu(221),d2fu(221)
      common/temp/ thk,htc,dudt,Cp,dens,tam,g0,ncell,lht,qq
      dimension terms(221),tt(200),cu(200),mc(200),tot(200)
   44 format(/'           mass  = ',f7.4,' kg/m2')
   45 format(' specific energy  = ',f8.2,' W-h/kg')
   46 format(' specific power   = ',f8.2,' W/kg')
c
      open(3,file='halfcells',status='unknown')
      write (3,*) 'time      negative       positive'
      print *, 'Enter input file name, press return'
      read *, filin
c       open (1, FILE = 'input.in', status = 'old')
      open (1, FILE = filin, status = 'old')
      print *, 'Enter output file name, press return'
      read *, filout
c       open (2, file = 'output.out', status = 'unknown')
      open (2, file = filout, status = 'unknown')
c
c     n is number of equations
      n=6
      lim2=20

      data fc/96487.0d0/, r/8.314d0/, pi/3.141592653589d0/
      data ed/0/, Vold/0/
c
c*****************************************************************
c     read in parameters and boundary conditions
c
      read (1,*) lim !limit on number of iterations
      read (1,*) h1  !thickness of negative electrode (m)
      read (1,*) h2  !thickness of separator (m)
      read (1,*) h3  !thickness of positive electrode (m)
      read (1,*) hcn !thickness of negative electrode current collector (m)
      read (1,*) hcp !thickness of positive electrode current collector (m)
      thk=h1+h2+h3
      read (1,*) n1  !number of nodes in negative electrode
c     If negative electrode is metal foil, let n1 = 0
      read (1,*) n2  !number of nodes in separator
      read (1,*) n3  !number of nodes in positive electrode
      read (1,*) t   !temperature (K)
      write (2, 1101) lim,1.d6*h1,1.d6*h2,1.d6*h3,1.d6*hcn,1.d6*hcp
     &,n1,n2,n3,t
      n2=n2+1
      nj=n1+n2+n3
c
      read (1,*) xi(1,n1+2)!initial concentration (mol/m3)
c     guess for PHI2
      xi(2,1)=0.05d0
      xi(2,nj)=0.0d0
      read (1,*) csx  !initial stochiometric parameter for negative
      read (1,*) csy  !initial stochiometric parameter for positive
      read (1,*) tmmax!maximum time step size (s)
      read (1,*) vcut !cutoff potential
      read (1,*) dfs1 !diffusion coefficient in negative solid (m2/s)
      read (1,*) dfs3 !diffusion coefficient in positive solid (m2/s)
      read (1,*) Rad1 !radius of negative particles (m)
c     If negative electrode is metal foil, let Rad1 = 1.0
      read (1,*) Rad3 !radius of positive particles (m)
      write (2,1102) xi(1,n1+2),csx,csy,tmmax,vcut,dfs1,dfs3,
     &1.d6*Rad1 ,1.d6*Rad3
c     If negative electrode is metal foil, let ep1=epp1=epf1=0.0
      read (1,*) ep1 !volume fraction of electrolyte in negative electrode
      read (1,*) epp1!volume fraction of polymer phase in negative electrode
      read (1,*) epf1!volume fraction of inert filler in negative electrode
      read (1,*) ep2 !volume fraction of electrolyte in separator
      read (1,*) epp2!volume fraction of polymer phase in separator
      read (1,*) ep3 !volume fraction of electrolyte in positive electrode
      read (1,*) epp3!volume fraction of polymer phase in positive electrode
      read (1,*) epf3!volume fraction of inert filler in positive electrode
      read (1,*) sig1!conductivity of solid negative matrix (S/m)
      read (1,*) sig3!conductivity of solid positive matrix (S/m)
c     read (1,*) cmax!maximum concentration in electrolyte          (mol/m3)
      read (1,*) rka1!reaction rate constant for negative reaction
      read (1,*) rka3!reaction rate constant for positive reaction
      read (1,*) ranode  !anode film resistance (out of place)
      read (1,*) rcathde !cathode film resistance (out of place)
c     read (1,*) il4 !1 for polymer, 0 for liquid electrolyte
      read (1,*) cot1 !coulombic capacity of negative material (mAh/g)
      read (1,*) cot3 !coulombic capacity of positive material (mAh/g)
      write (2,1103) ep1,epp1,epf1,ep2,epp2,ep3,epp3,epf3,sig1,
     & sig3,cot1, cot3,cmax,rka1,rka3,il4
      read (1,*) re  ! density of electrolyte (kg/m3)
      read (1,*) rs1 ! density of negative insertion material (kg/m3)
      read (1,*) rs3 ! density of positive insertion material (kg/m3)
      read (1,*) rf  ! density of inert filler (kg/m3)
      read (1,*) rpl ! density of polymer phase (kg/m3)
      read (1,*) rc  ! density of separator material (kg/m3)
      read (1,*) rcn ! density of negative current collector (kg/m3)
      read (1,*) rcp ! density of positive current collector (kg/m3)
      write (2,1104) re,rs1,rs3,rf,rpl,rc,rcn,rcp
      read (1,*) htc  !heat transfer coefficient with external medium (W/m2K)
      read (1,*) dUdT !temperature coefficient of EMF (V/K)
      read (1,*) Cp   !heat capacity of cell (J/kgK)
      read (1,*) Tam  !ambient temperature (K)
      read (1,*) ncell!number of cells in a cell stack
      read (1,*) lht  !0 uses htc, 1 calcs htc, 2 isothermal
      write (2,1105) ranode,rcathde,htc,dudt,Cp,tam,ncell,lht
      read (1,*) il1   !1 for long print-out 0 for short print-out
      read (1,*) il2   !1/il2 = fraction of nodes in long print-out
      read (1,*) il3   !1/il3 = fraction of time steps in long print-out
      read (1,*) lflag ! 0 for electrolyte in separator only, 1 for uniform
      read (1,*) lpow  ! 0 for no power peaks, 1 for power peaks
      read (1,*) jsol  ! calculate solid profiles if 1<jsol<j< j < n2+n1 ]
c      in separator
c
      h=h2
c
      b(5,4)=1.0d0
      g(5)=cur-c(4,j)
c
      b(6,6)=1.0d0
      g(6)=-c(6,j)
c
c      do 504 i=3,3
      call band(j)
      go to 10
c      _____________________________________________________
  130 if (j .ne. (n2+n1)) go to 140
c      Boundary between positive and separator(j=n2+n1):
c
      if(cap3.eq.0.d0) then
      b(5,4)=-1.0d0/h
      a(5,4)=1.0d0/h
      b(5,5)=area3*fc/2.0d0
      g(5)=(c(4,j)-c(4,j-1))/h-area3*fc/2.0d0*c(5,j) ! not order h2
      else
      if (rr .eq. 0) then
      g(5)=c(6,j)-c(2,j)-xt(6,j,kk-1)+xt(2,j,kk-1)
      b(5,6)=-1.d0
      b(5,2)= 1.d0
      else
      b(5,4)=-1.0d0/h/2.0d0
      a(5,4)=1.0d0/h/2.0d0
      b(5,5)=area3*fc/4.0d0
      b(5,6)=area3*cap3/rr*0.5d0
      b(5,2)=-area3*cap3/rr*0.5d0
      g(5)=(c(4,j)-c(4,j-1)+xt(4,j,kk-1)-xt(4,j-1,kk-1))/h/2.0d0
     &    -area3*fc/4.0d0*(c(5,j)+xt(5,j,kk-1))
     &    -area3*cap3*(c(6,j)-c(2,j)-xt(6,j,kk-1)
     &    +xt(2,j,kk-1))/rr*0.5d0
      endif
      endif
c
      d(6,6)=1.0d0/h3
      b(6,6)=-1.0d0/h3
      b(6,4)=-1.0d0/sig3
      g(6)=-cur/sig3+c(4,j)/sig3-(c(6,j+1)-c(6,j))/h3
c
c      do 505 i=3,3
      call band(j)
      go to 10
c      ____________________________________________________________
  140 if (j .eq. nj) go to 16
c
c      specify governing equations [ n2+n1 < j < nj ]
c      composite cathode
c
      h=h3
c
      if(cap3.eq.0.d0) then
      b(5,4)=-1.0d0/h
      a(5,4)=1.0d0/h
      b(5,5)=area3*fc
      g(5)=(c(4,j)-c(4,j-1))/h-area3*fc*c(5,j)
      else
      if (rr .eq. 0) then
      g(5)=c(6,j)-c(2,j)-xt(6,j,kk-1)+xt(2,j,kk-1)
      b(5,6)=-1.d0
      b(5,2)= 1.d0
      else
      b(5,4)=-1.0d0/h/2.0d0
      a(5,4)=1.0d0/h/2.0d0
      b(5,5)=area3*fc/2.0d0
      b(5,6)=area3*cap3/rr
      b(5,2)=-area3*cap3/rr
      g(5)=(c(4,j)-c(4,j-1)+xt(4,j,kk-1)-xt(4,j-1,kk-1))/h/2.0d0
     &    -area3*fc/2.0d0*(c(5,j)+xt(5,j,kk-1))
     &    -area3*cap3*(c(6,j)-c(2,j)-xt(6,j,kk-1)
     &    +xt(2,j,kk-1))/rr
      endif
      endif
c
      d(6,6)=1.0d0/h
      b(6,6)=-1.0d0/h
      b(6,4)=-1.0d0/sig3
      g(6)=-cur/sig3+c(4,j)/sig3-(c(6,j+1)-c(6,j))/h3
c
c      do 506 i=3,3
      call band(j)
      go to 10
c
c     ________________________________________________
c
   16 continue
c     specify boundary conditions at right interface(j=nj)
c
      if(cap3.eq.0.d0) then
      b(5,4)=-1.0d0/h
      a(5,4)=1.0d0/h
      b(5,5)=area3*fc/2.0d0
      g(5)=(c(4,j)-c(4,j-1))/h-area3*fc/2.0d0*c(5,j) ! not order h2
      else
      if (rr .eq. 0) then
      g(5)=c(6,j)-c(2,j)-xt(6,j,kk-1)+xt(2,j,kk-1)
      b(5,6)=-1.d0
      b(5,2)= 1.d0
      else
      b(5,4)=-1.0d0/h/2.0d0
      a(5,4)=1.0d0/h/2.0d0
      b(5,5)=area3*fc/4.0d0
      b(5,6)=area3*cap3/rr*0.5d0
      b(5,2)=-area3*cap3/rr*0.5d0
      g(5)=(c(4,j)-c(4,j-1)+xt(4,j,kk-1)-xt(4,j-1,kk-1))/h/2.0d0
     &    -area3*fc/4.0d0*(c(5,j)+xt(5,j,kk-1))
     &    -area3*cap3*(c(6,j)-c(2,j)-xt(6,j,kk-1)
     &    +xt(2,j,kk-1))/rr*0.5d0
      endif
      endif
c
      b(6,4)=1.0d0
      g(6)=-c(4,j) !  i2 is no longer used at j=nj
c
```

```
c     do 507 i=3,3
      call band(j)
      do 607 jj=1,nj
      do 607 i=1,n
  607 c(i,jj)=xx(i,jj)+c(i,jj)
c     ______________________________________________________
c
c     check for convergence
c
      do 56 i=1,n
   56 xp(i)=(4.0d0*c(i,2)-3.0d0*c(i,1)-c(i,3))/2.0d0/h1
c
      nerr=0
      do 25 j=1,nj
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
c     shoe horns:
      if(c(1,j).lt.xx(1,j)/100.) c(1,j)=xx(1,j)/100.
      if (c(2,j).lt.(xx(2,j)-0.02)) c(2,j)=xx(2,j)-0.02
      if (c(2,j).gt.(xx(2,j)+0.02)) c(2,j)=xx(2,j)+0.02
      if (c(6,j).lt.(xx(6,j)-0.02)) c(6,j)=xx(6,j)-0.02
      if (c(6,j).gt.(xx(6,j)+0.02)) c(6,j)=xx(6,j)+0.02
      if (c(2,j).gt. 9.9) c(2,j)= 9.9
      if (c(2,j).lt.-9.9) c(2,j)=-9.9
      if (c(6,j).gt. 9.9) c(6,j)= 9.9
      if (c(6,j).lt.-9.9) c(6,j)=-9.9
c
      if (j .ge. n1+n2) then
      if(c(3,j).lt.xx(3,j)/100.) nerr=nerr+1
      if(c(3,j).lt.xx(3,j)/100.) c(3,j)=xx(3,j)/100. ! use cs min
      if(ct3-c(3,j).le.(ct3-xx(3,j))/100.) nerr=nerr+1
      if(ct3-c(3,j).le.(ct3-xx(3,j))/100.) c(3,j)=ct3-(ct3-xx(3,j))/100.
      if(c(3,j).ge.ct3) c(3,j)=0.999999*ct3
      if(c(3,j).lt.1.0d-12) c(3,j)=1.0d-12
c
      else if (j .le. n1+1 .and. n1 .gt. 0) then
      if(c(3,j).lt.xx(3,j)/100.) nerr=nerr+1
      if(c(3,j).lt.xx(3,j)/100.) c(3,j)=xx(3,j)/100. ! use cs min
      if(ct1-c(3,j).le.(ct1-xx(3,j))/100.) nerr=nerr+1
      if(ct1-c(3,j).le.(ct1-xx(3,j))/100.) c(3,j)=ct1-(ct1-xx(3,j))/100.
      if(c(3,j).ge.ct1) c(3,j)=0.999999*ct1
      if(c(3,j).lt.1.0d-99) c(3,j)=1.0d-99
      endif
c     to avoid underflow or overflow:
      if(c(1,j).lt.1.0d-12) c(1,j)=1.0d-12
      if(c(1,j).lt.1.0d-10) c(5,j)=0.0
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
c
      do 25 i=1,n
   25 xx(i,j)=c(i,j)
c
      if (jcount .gt. 3*lim .and. rr.eq.0.0d0) then
      write (2, 99)
      stop
      endif
c      ________________________________________________
c
c      Decreasing time steps:
      if (jcount .gt. lim .and. rr.gt.0.0d0) then
      tau=tau/2.0d0
      rr=tau
      ts(kk)=ts(kk-1)+tau
      write (2,*) 'time step reduced to ', tau, ts(kk)
      if (tau.lt.1.0d-4) then
      if (lpow.eq.1) then  !peak power activated
      nflag=1
      go to 66
      endif
      nt=nt-1
      tau=(ts(kk-1)-ts(kk-2))/2.d0
      rr=tau
      kback=0
      ed=ed/tw/3.6d03
      pow=3.6d03*ed/ts(nt+1)
      write (2,*) 'mass is ',tw
      write (2,*) 'energy is ',ed
      write (2,*) 'power is ',pow
      write (2,*) kk-1,' this time step did not converge'
      call nucamb(1,5)
      stop
      else
      iflag=0
      call calca(kk)
      go to 666
      end if
c
      else
c
      if(nerr.ne.0) go to 8
      do 55 ii=1,n
      errlim=1.d-10
      if(ii.eq.5) errlim=1.d-16 !change to -14 if problems with convergence
      dxp=dabs( xp(ii)-xp0(ii) )
      if (n1 .lt. 11) then
      n1hold = 1
      else
      n1hold = n1-10
      endif
      dxx=dabs( xx(ii,n1hold) - xx0(ii,n1hold) )
      dxx2=dabs( xx(ii,n1+n2+10)-xx0(ii,n1+n2+10) )
      if(dxx .gt. 1.d-9*dabs(xx(ii,n1hold)).and.dxx.gt.errlim) go to 8
      if(dxx2.gt.1.d-9*dabs(xx(ii,n1+n2+10)).and.dxx2.gt.errlim)
     & go to 8
c      if(dxp.gt.1.d-7*dabs(xp(ii)) .and. dxp.gt.errlim) go to 8
   55 continue
c
c      ________________________________________________
c
      if(lpow.ne.1) write (2,*) jcount,' iterations required'
c
      do 60 ll=1, nj  ! save present time results in xt()
      do 60 lk=1,n
   60 xt(lk,ll,kk)=xx(lk,ll)
c
c      do 57 j=1,nj
c      if(xx(1,j) .lt. 1.0d-03) fj(j)=1
c  57 if(xx(1,j) .gt. 1.0d-01) fj(j)=0
c
c- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      if(rr.ne.0.0d0) then
      do 58 j=1,nj  ! fix to calculate here for zero time step
   58 term(j)=termn(j)
      else
      do 65 j=1,nj
      term(j)=0.
      fac=1.
      if(j.eq.n1+2 .and. n1 .gt. 0)
     &     fac=((ep2+epp2)/(ep1+epp1))**exbrug
      if(j.eq.n1+n2+1) fac=((ep3+epp3)/(ep2+epp2))**exbrug
      epn=ep1+epp1
      hn=h1
      if(j.gt.n1+1) then
      epn=ep2+epp2
      hn=h2
      endif
      if (j.gt.n1+n2) then
      epn=ep3+epp3
      hn=h3
      endif
      if(j.gt.1) term(j)=
     &-(df(j)+fac*df(j-1))*(c(1,j)-c(1,j-1))/hn/2.
     &-(1.-0.5*(tm(j)+tm(j-1)))*c(4,j-1)/fc
      fac=1.
      if(j.eq.n1+1) then
      if (n1 .gt. 0) fac=((ep2+epp2)/(ep1+epp1))**exbrug
      epn=ep2+epp2
      hn=h2
      else if(j.eq.n1+n2) then
      fac=((ep3+epp3)/(ep2+epp2))**exbrug
      epn=ep3+epp3
      hn=h3
      endif
   65 if(j.lt.nj) term(j)=term(j)
     &-(fac*df(j)+df(j+1))*(c(1,j)-c(1,j+1))/hn/2.
     &+(1.-0.5*(tm(j)+tm(j+1)))*c(4,j)/fc
      endif
c
c
      end if
c
   66 continue
      return
      end
c
c***********************************************************************
      subroutine calca(kk)
      implicit real*8(a-h,o-z)
      parameter(maxt=900)
      common /n/ nx,nt,n1,n2,nj,n3,tmmax
      common /calc/ ai(maxt),ai2(maxt),u(223,maxt),ts(maxt),
     1h,h1,h2,h3,hcn,hcp,rr,rrmax
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     1ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1
      common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,
     1sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
      dimension ar(4,maxt),bz(6)
c
      do 319 l=1,nt
      ai2(l)=0.0d0
  319 ai(l)=0.0d0
c
```

```
      do 70 i=1,kk-1
      ar(1,i)=dfs3*(ts(kk)-ts(i))/Rad3/Rad3
      ar(2,i)=dfs3*(ts(kk)-ts(i+1))/Rad3/Rad3
      ar(3,i)=dfs1*(ts(kk)-ts(i))/Rad1/Rad1
      ar(4,i)=dfs1*(ts(kk)-ts(i+1))/Rad1/Rad1
c
      do 69 m=1,2
      t1=ar(m,i)
      t2=ar((m+2),i)
c
      a1=0.0d0
      a12=0.0d0
c
      s=1.644934066848d0
c
c      Bessel's function zeros:
      bz(1)=2.4048255577d0
      bz(2)=5.5200781103d0
      bz(3)=8.6537281103d0
      bz(4)=11.7915344391d0
      bz(5)=14.9309177086d0
c
      if (shape3.gt.2.0d0) then
c      spherical particles:
      if (t1 .gt. 0.06d0) then
c
      do 59 j=1,5
      y1=j*j*pi*pi*t1
   59 if (y1 .le. 1.5d02) a1=a1+(expf(-y1))/j/j
      a1=2.0d0*(s-a1)/pi/pi
c
      else
c
      if (t1.LE.0.0d0) then
      a1=0.0d0
      else
      do 60 j=1,3
      z=j/dsqrt(t1)
      call erfc(z,e)
      y2=j*j/t1
      if(y2 .ge. 1.5d02) then
      da=-j*dsqrt(pi/t1)*e
      else
      da=expf(-y2)-j*dsqrt(pi/t1)*e
      end if
   60 a1=a1+da
```

```
      a1=-t1 + 2.0d0*dsqrt(t1/pi)*(1.0d0+2.0d0*a1)
      end if
c
      end if
      else
c
      if (shape3.lt.2.0d0) then
c     planar particles:
      if(t1 .gt. 0.06d0) then
c
      do 61 j=1,5
      da=((-1.0d0)**(j))*(1.0d0 - expf(-(2.0d0*j+1.0d0)*
     1(2.0d0*j+1.0d0)*pi*pi*t1))/(2.0d0*j+1.0d0)/(2.0d0*j+1.0d0)
   61 a1=a1+da
      a1=4.0d0*a1/pi/pi
c
      else
c
      do 62 j=1,3
      z=j/2.0d0/dsqrt(t1)
      call erfc(z,e)
      da=((-1.0d0)**(j))*(expf(-j*j/4.0d0/t1)-j/2.0d0*dsqrt(pi/t1)*e)
   62 a1=a1+da
      a1=2.0d0*dsqrt(t1/pi)*(1.0d0+2.0d0*a1)
c
      end if
      else
c     cylindrical particles:
      if (t1.gt.0.06d0) then
c
      do 63 j=1,5
      da=(1.0d0-expf(-bz(j)*bz(j)*t1))/bz(j)/bz(j)
   63 a1=a1+da
      a1=2.0d0*a1
c
      else
c
      a1=2.0d0*dsqrt(t1/pi)-t1/4.0d0-5.0d0*(t1**1.5d0)/96.0d0/dsqrt(pi)
     1-31.0d0*t1*t1/2048.0d0
c
      end if
      end if
      end if
c
      if (n1 .eq. 0) go to 36
c     (skip calculations of Li diffusion in the solid
c    if the negative electrode is metal foil)
c
      if (shape1.gt.2.0d0) then
c     spherical particles:
      if(t2 .gt. 0.06d0) then
c
      do 64 j=1,5
      y2=j*j*pi*pi*t2
 64   if(y2 .le. 1.5d02) a12=a12+(expf(-y2))/j/j
      a12=2.0d0*(s-a12)/pi/pi
c
      else
c
      if (t2.eq.0.0d0) then
      a12=0.0d0
      else
      do 65 j=1,3
      z=j/dsqrt(t2)
      call erfc(z,e)
      y2=j*j/t2
      if(y2 .gt. 1.5d02) then
      da=-j*dsqrt(pi/t2)*e
      else
      da=expf(-y2)-j*dsqrt(pi/t2)*e
      end if
 65   a12=a12+da
      a12=-t2 + 2.0d0*dsqrt(t2/pi)*(1.0d0+2.0d0*a12)
      end if
      end if
c
      else
      if (shape1.lt.2.0d0) then
c     planar particles:
      if(t2 .gt. 0.06d0) then
c
      do 66 j=1,5
      da=((-1.0d0)**(j))*(1.0d0 - expf(-(2.0d0*j+1.0d0)*
     1(2.0d0*j+1.0d0)*pi*pi*t2))/(2.0d0*j+1.0d0)/(2.0d0*j+1.0d0)
 66   a12=a12+da
      a12=4.0d0*a12/pi/pi
c
      else
c
      do 67 j=1,3
      z=j/2.0d0/dsqrt(t2)
      call erfc(z,e)
      da=((-1.0d0)**(j))*(expf(-j*j/4.0d0/t2)-j/2.0d0*dsqrt(pi/t2)*e)
 67   a12=a12+da
      a12=2.0d0*dsqrt(t2/pi)*(1.0d0+2.0d0*a12)
c
      end if
      else
c     cylindrical particles:
      if (t2.gt.0.06d0) then
c
      do 68 j=1,5
      da=(1.0d0-expf(-bz(j)*bz(j)*t2))/bz(j)/bz(j)
 68   a12=a12+da
      a12=2.0d0*a12
c
      else
c
      a12=2.0d0*dsqrt(t2/pi)-t2/4.0d0-5.0d0*(t2**1.5d0)/96.0d0/dsqrt(pi)
     1-31.0d0*t2*t2/2048.0d0
c
      end if
      end if
      end if
c
 36   continue
c
      ar(m,i)=a1
 69   ar((m+2),i)=a12
c
      ai(kk-i)=ar(1,i)-ar(2,i)
 70   ai2(kk-i)=ar(3,i)-ar(4,i)
c
      return
      end
c********************************************************************
      subroutine erfc(z,e)
      implicit real*8(a-h,o-z)
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     1ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1
c
      a1=0.254829592d0
      a2=-0.284496736d0
      a3=1.421413741d0
      a4=-1.453152027d0
      a5=1.061405429d0
      if(z .lt. 2.747192d0) then
      t3=1.0d0/(1.0d0+0.3275911d0*z)
      e=(a1*t3+a2*t3*t3+a3*(t3**3.0d0)+a4*(t3**4.0d0)
     1+a5*(t3**5.0d0))*expf(-z*z)
      else
c
      if(z .gt. 25.0d0) then
      e=0.0d0
      else
c
      sum=0.0d0
      max=z*z + 0.5
      fac=-0.5d0/z/z
      sum=fac
      tl=fac
      n=1
   10 n=n+1
      if(n .gt. max) go to 15
      tn=tl*(2.0d0*n-1.0d0)*fac
      sum=sum + tn
      if(tn .lt. 1.0d-06) go to 15
      tl=tn
      go to 10
   15 e=(expf(-z*z))*(1.0d0+sum)/dsqrt(pi)/z
      end if
      end if
c
      return
      end
c**********************************************************************
      subroutine band(j)
      implicit real*8(a-h,o-z)
      common /n/ nx,nt,n1,n2,nj,n3,tmmax
      common/mat/ b,d
      common/bnd/ a,c,g,x,y
      dimension b(10,10),d(10,21)
      dimension a(10,10),c(10,221),g(10),x(10,10),y(10,10)
      dimension e(10,11,221)
  101 format (15h determ=0 at j=,i4)
      n=nx
      if (j-2)  1,6,8
    1 np1= n + 1
      do 2 i=1,n
      d(i,2*n+1)= g(i)
      do 2 l=1,n
      lpn= l + n
    2 d(i,lpn)= x(i,l)
      call matinv(n,2*n+1,determ)
    if (determ)  4,3,4
  3 write (2, 101) j
  4 do 5 k=1,n
    e(k,np1,1)= d(k,2*n+1)
    do 5 l=1,n
    e(k,l,1)= - d(k,l)
    lpn= l + n
  5 x(k,l)= - d(k,lpn)
    return
  6 do 7 i=1,n
    do 7 k=1,n
    do 7 l=1,n
  7 d(i,k)= d(i,k) + a(i,l)*x(l,k)
  8 if (j-nj)  11,9,9
  9 do 10 i=1,n
    do 10 l=1,n
    g(i)= g(i) - y(i,l)*e(l,np1,j-2)
    do 10 m=1,n
  10 a(i,l)= a(i,l) + y(i,m)*e(m,l,j-2)
  11 do 12 i=1,n
    d(i,np1)= - g(i)
    do 12 l=1,n
    d(i,np1)= d(i,np1) + a(i,l)*e(l,np1,j-1)
    do 12 k=1,n
  12 b(i,k)= b(i,k) + a(i,l)*e(l,k,j-1)
    call matinv(n,np1,determ)
    if (determ)  14,13,14
  13 write (2, 101) j
  14 do 15 k=1,n
    do 15 m=1,np1
  15 e(k,m,j)= - d(k,m)
    if (j-nj)  20,16,16
  16 do 17 k=1,n
  17 c(k,j)= e(k,np1,j)
    do 18 jj=2,nj
    m= nj - jj + 1
    do 18 k=1,n
    c(k,m)= e(k,np1,m)
    do 18 l=1,n
  18 c(k,m)= c(k,m) + e(k,l,m)*c(l,m+1)
    do 19 l=1,n
    do 19 k=1,n
  19 c(k,1)= c(k,1) + x(k,l)*c(l,3)
  20 return
    end
c*******************************************************************
```

```
  subroutine matinv(n,m,determ)
  implicit real*8(a-h,o-z)
  common/mat/ b,d
  dimension b(10,10),d(10,21)
  dimension id(10)
  determ=1.0
  do 1 i=1,n
1 id(i)=0
  do 18 nn=1,n
  bmax=1.1
  do 6 i=1,n
  if(id(i).ne.0) go to 6
  bnext=0.0
  btry=0.0
  do 5 j=1,n
  if(id(j).ne.0) go to 5
  if(dabs(b(i,j)).le.bnext) go to 5
  bnext=dabs(b(i,j))
  if(bnext.le.btry) go to 5
  bnext=btry
  btry=dabs(b(i,j))
  jc=j
5 continue
  if(bnext.ge.bmax*btry) go to 6
  bmax=bnext/btry
  irow=i
  jcol=jc
6 continue
  if(id(jc).eq.0) go to 8
  determ=0.0
  return
8 id(jcol)=1
  if(jcol.eq.irow) go to 12
  do 10 j=1,n
  save=b(irow,j)
  b(irow,j)=b(jcol,j)
10 b(jcol,j)=save
  do 11 k=1,m
  save=d(irow,k)
  d(irow,k)=d(jcol,k)
11 d(jcol,k)=save
12 f=1.0/b(jcol,jcol)
  do 13 j=1,n
13 b(jcol,j)=b(jcol,j)*f
  do 14 k=1,m
14 d(jcol,k)=d(jcol,k)*f
```

```
      do 18 i=1,n
      if(i.eq.jcol) go to 18
      f=b(i,jcol)
      do 16 j=1,n
   16 b(i,j)=b(i,j)-f*b(jcol,j)
      do 17 k=1,m
   17 d(i,k)=d(i,k)-f*d(jcol,k)
   18 continue
      return
      end
c*******************************************************************
      subroutine nucamb(il2,il3)
      implicit real*8(a-h,o-z)
      parameter(maxt=900)
      common /n/ nx,nt,n1,n2,nj,n3,tmmax
      common /calc/ ai(maxt),ai2(maxt),u(223,maxt),ts(maxt),
     1h,h1,h2,h3,hcn,hcp,rr,rrmax
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     1ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1
      common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
      common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,
     1sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
      common/tprop/df(221),cd(221),tm(221),
     1ddf(221),dcd(221),dtm(221),dfu(221),d2fu(221)
      dimension zz(221)
  109 format(f6.1,', ',f15.5,', ',f7.4,', 'g10.4,', ',f6.2,', ',g10.4
     1,', ',f10.4)
  309 format(f8.5,', ',f8.5)
   44 format(' t = ',1pe18.6,' min')
c
      do 5 i=1,n1+1
      w=i-1
    5 zz(i) = w*h1*1.0d06
      do 71 i=n1+2,n2+n1
      w=i-(n1+1)
   71 zz(i)=zz(n1+1)+w*h2*1.0d06
      do 72 i=n1+n2+1,nj
      w=i-(n1+n2)
   72 zz(i)=zz(n2+n1)+w*h3*1.0d06
c
      do 11 l=1,nt+1
      if(ts(l)-ts(l-1).lt.1.d-6) go to 9
      if (l.lt.nt-5 .and. mod(l-1,il3).ne.0) go to 11
    9 write (2,*) ' '
      write (2,*) 'distance concen      PHI2    c solid',
     1'    current      j       PHI1'
```

```
      write (2,*) 'microns  (mol/m3)   (V)    x or y ',
     1'    (A/m2)  (A/m3)      (V)'
      write (2,44) ts(l)/60.0d0
c
      do 10 j=1,nj,il2
      if (j .le. n1+1) then
      csol=ct1
      else
      csol=ct3
      end if
      if(j.le.n1+1) then
      curden=area1*fc*xt(5,j,l)
      else if(j.ge.n1+n2) then
      curden=area3*fc*xt(5,j,l)
      else
      curden=0.0
      endif
   10 write(2,109) zz(j),xt(1,j,l),xt(2,j,l),xt(3,j,l)/csol,xt(4,j,l)
     1,curden,xt(6,j,l)
   11 continue
c
      return
      end
c*********************************************************************
      subroutine guess(lflag)
      implicit real*8(a-h,o-z)
      parameter(maxt=900)
      common /n/ nx,nt,n1,n2,nj,n3,tmmax
      common /calc/ ai(maxt),ai2(maxt),u(223,maxt),ts(maxt),
     1h,h1,h2,h3,hcn,hcp,rr,rrmax
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     1ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1
      common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
      common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,
     1sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
      common/tprop/df(221),cd(221),tm(221),
     1ddf(221),dcd(221),dtm(221),dfu(221),d2fu(221)
      dimension del(6)
c
      del(2)=cur*h/2.5
      if (n1 .gt. 1) then
      del(3)=cur/(n1)
      else
      del(3) = cur
      endif
      del(4)=cur/(n3)
      del(5)=(xi(5,nj)-xi(5,1))/(nj-1)
c
      Ua=xi(6,1)
      Uc=xi(6,nj)
      do 73 i=1,(n1+1)
      xi(2,i)=xi(2,1)*(nj-i)/nj
      xi(3,i)=xi(3,1)
      xi(4,i)=xi(4,1)+del(3)*(i-1)
      xi(5,i)=xi(5,1)
   73 xi(6,i)=xi(2,i)+Ua
c
      do 74 i=(n1+2),(n2+n1-1)
c      xi(2,i)=xi(2,1)-del(2)*(i-n1-2)
      xi(2,i)=xi(2,1)*(nj-i)/nj
      xi(3,i)=0.0d0
      xi(4,i)=cur
      xi(5,i)=0.0d0
   74 xi(6,i)=0.0d0
c
      do 75 i=(n2+n1),nj
      xi(2,i)=xi(2,1)*(nj-i)/nj
      xi(3,i)=xi(3,nj)
      xi(4,i)=xi(4,n2+n1)-del(4)*(i-n1-n2)
      xi(5,i)=xi(5,n2+n1)
   75 xi(6,i)=xi(2,i)+Uc
c
      do 15 i=1,nj
      xt(6,i,1)=xi(6,i)
      xt(5,i,1)=xi(5,i)
      xt(4,i,1)=xi(4,i)
      xt(3,i,1)=xi(3,i)
   15 xt(2,i,1)=xi(2,i)
c
      do 16 i=1,nj
      xi(1,i)=xi(1,n1+2)
c      Uniform initial concentration if lflag=1
c      Step function initial concentration if lflag=0
      if(lflag.eq.0 .and. (i.le.n1+1 .or. i.ge.n1+n2))
     ξ(1,i)=1.0d-01
   16 xt(1,i,1)=xi(1,i)
      return
      end
c**********************************************************************
      subroutine peak(n,lim,curr)
      implicit real*8(a-h,o-z)
      parameter(maxt=900)
      common /n/ nx,nt,n1,n2,nj,n3,tmmax
      common /calc/ ai(maxt),ai2(maxt),u(223,maxt),ts(maxt),
     1h,h1,h2,h3,hcn,hcp,rr,rrmax
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     1ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1
      common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
      common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,
     1sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
  311 format(f8.5,', ',f7.3,',  ',f8.3,', ',f8.5)
c
c     Peak power current ramp section:
      write (2,*) ' '
      write (2,*) '    PEAK POWER '
      write (2,*) ' '
      write (2,*) 'cell pot  ',' current','   power   min pot'
      write (2,*) '  (V)  ','    (A/m2)','   (W/m2)     (V)'
c
c     Duration of current pulse is 30 seconds.
c
      vcut=2.8d0
      curmin=0.d0
      pwrpmax=0.d0
      rrmax=30.0d0
      cur=curr
  127 kcount=0
      fact=20.0d0
      curmax=0.d0
      vfmax=0.d0
      k=nt+2
      do 126 j=1,nj
      do 126 i=1,n
  126 xt(i,j,k)=xt(i,j,k-1)
      ppow=0.0
      ii=0
c     Ramp current:
c
  128 continue
      if (ii.gt.60) return
c
      energ=0.0
  130 ii=ii+1
      cur=cur+fact  ! start a new current density
        if(curmax.ne.0.d0) then
        cur=0.5d0*(curpmax+curmax)
        if(mod(ii,2).eq.1) cur=0.5d0*(curpmax+curmin)
        if(cur.eq.0.d0) cur=0.5d0*(curmax+curmin)
        if(vfmax.gt.0.d0 .and. vfmax.lt.vcut) then
        curtry=curpmax+(vcut-vvpmax)/(vfmax-vvpmax)*(curmax-curpmax)
     &+0.01d0*(curmax-curpmax)*dble(mod(ii,3)-1)
        if(curtry.lt.curmax .and. curtry.gt.curpmax) cur=curtry
        endif
      if(vfmax.gt.vcut .and. curmin.gt.0.0) then
      v2=pwrmax/curmax
      cur2=curmax
      v1=pwrmin/curmin
      cur1=curmin
      vm=pwrpmax/curpmax
      curm=curpmax
      resis=-((pwrpmax-pwrmin)/(curpmax-curmin)
     &-(pwrpmax-pwrmax)/(curpmax-curmax))/(curmin-curmax)
      Uop=(pwrpmax-pwrmin)/(curpmax-curmin)+resis*(curpmax+curmin)
      curtry=Uop/2.d0/resis+0.1d0*(curmax-curmin)*dble(mod(ii,3)-1)
      write (2,*) 'curtry= ',curtry,resis,Uop
      if(curtry.lt.curmax .and. curtry.gt.curmin) cur=curtry
      endif
        endif
      write (2,*) ' cur= ',cur,curmin,curpmax,curmax
      kkflag=0
      iflag=0
      nflag=0
      k=nt+2
      timpk=0.0d0
      rr=0.0d0
      ts(k)=ts(k-1)
      call comp(n,lim,k,rr,kkflag,nflag,1,jcount)
      call cellpot(k,vv,0,1,lflag)
      vlast=vv
      rr=0.2d0
  129 kkflag=kkflag+1
      k=k+1
      ts(k)=ts(k-1)+rr
      call calca(k)
c
      call comp(n,lim,k,rr,kkflag,nflag,1,jcount)
c
      if (nflag.eq.1.and.kcount.lt.20) then
      write (2,*) 'Peak current decreased',kcount,fact
      if(cur.lt.curpmax) then
      write (2,*) 'Convergence on power failed; already converged at a high
     &er current'
      write (2,*) 'Best results obtained are:'
      write (2, 311) pwrpmax/curpmax,curpmax,pwrpmax,vvpmax
      write (2,*) ' finished with vcut= ',vcut
      return
      endif
      curmax=cur
      vfmax=0.d0
      go to 128
      endif
      if (kcount.ge.10) return
c
      call cellpot(k,vv,0,1,lflag)
      energ=energ+(vlast+vv)*(ts(k)-ts(k-1))*cur/2.0d0
c
      timpk=timpk+rr
      if (dabs(timpk-30.0d0).gt.0.1) then
c
      if (timpk.lt.30.0) then
      vlast=vv
c      Increasing time steps:
      if(jcount.lt.6 .and. kkflag.gt.5 .and. (2.0d0*rr
     1+timpk).lt.30.0d0 .and. iflag.eq.0) then
        rr=rr*2.0d0
c         write (2,*) 'next time step increased to ', rr,'(s)'
      end if
      if(timpk+rr.gt.30.0) iflag=1
      if(timpk+rr.gt.30.0) rr=30.0d0-timpk
      go to 129
      end if
c
      end if
      ppow=energ/30.0d0
      write (2, 311) ppow/cur,cur,ppow,vv
        if(ppow.gt.pwrpmax .and. vv.gt.vcut) then
        if(curpmax.lt.cur) then
          curmin=curpmax
          pwrmin=pwrpmax
          vfmin=vvpmax
        else
          curmax=curpmax
          pwrmax=pwrpmax
          vfmax=vvpmax
        endif
        curpmax=cur
        pwrpmax=ppow
        vvpmax=vv
        endif
        if(vv.lt.vcut) then
        if(curmax.eq.0.d0 .or. cur.lt.curmax) then
          curmax=cur
          pwrmax=ppow
          vfmax=vv
        endif
        else
        if(cur.gt.curmin .and. cur.lt.curpmax) then
          curmin=cur
          pwrmin=ppow
          vfmin=vv
        endif
        if(cur.gt.curpmax) then
          curmax=cur
          pwrmax=ppow
          vfmax=vv
        endif
        endif
      if(curmax.eq.0.0d0) go to 128
      if(curmin. lt. 0.999d0*curmax) go to 128
      write (2, 311) pwrpmax/curpmax,curpmax,pwrpmax,vvpmax
      write (2, *) ' finished with vcut= ',vcut
      if(vcut.eq.0.0d0) return
      vcut=0.0d0
      go to 127
c
      end
c**********************************************************************
      subroutine cellpot(kk,v,li,lpow,lflag)
      implicit real*8(a-h,o-z)
      parameter(maxt=900)
      common /n/ nx,nt,n1,n2,nj,n3,tmmax
      common /calc/ ai(maxt),ai2(maxt),u(223,maxt),ts(maxt),
     1h,h1,h2,h3,hcn,hcp,rr,rrmax
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     1ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1
      common/power/ ed,Vold,ranode,rcathde
      common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
      common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,
     1sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
      common/temp/ thk,htc,dudt,Cp,dens,tam,g0,ncell,lht,qq
  309 format(f8.5,', ',f8.5,', ',f8.3,', ',g11.6,',',' ',f9.5,', ',f7.3,
     & ',',g10.5)
  307 format(f8.5,', ',f8.5,', ',f9.5,', ',g11.6,',','  ',f7.3,',',f8.3)
c
c      Material balance criteria:
      sum=0.0d0
```

```
       if (n1 .gt. 2) then
       do 85 j=2,n1
    85 sum=sum+xt(1,j,kk)*(ep1+epp1)*h1
       endif
       sum=sum+(xt(1,1,kk)+xt(1,n1+1,kk))*(ep1+epp1)*h1/2.0d0
       do 86 j=n1+2,n2+n1-1
    86 sum=sum+xt(1,j,kk)*(ep2+epp2)*h2
       sum=sum+(xt(1,n1+1,kk)+xt(1,n2+n1,kk))*(ep2+epp2)*h2/2.0d0
       do 87 j=n2+n1+1,nj-1
    87 sum=sum+xt(1,j,kk)*(ep3+epp3)*h3
       sum=sum+(xt(1,n1+n2,kk)+xt(1,nj,kk))*h3*(ep3+epp3)/2.0d0
c      calculate total salt in cell from initial profile:
       w=xt(1,n1+2,1)*((n2-1)*(ep2+epp2)*h2+n1*(ep1+epp1)*h1
      1+n3*(ep3+epp3)*h3)
       if(lflag.eq.0) w=w-(xt(1,n1+2,1)-xt(1,1,1))*(n1*(ep1+epp1)*h1
      1+n3*(ep3+epp3)*h3)
       if(lflag.eq.0) w=w-(xt(1,n1+2,1)-xt(1,1,1))*(ep2+epp2)*h2
c      material balance parameter should be ca=1.00
       ca=sum/w
c
       if (kk.eq.1) then
       ut=xt(3,nj,1)/ct3
       ut2=xt(3,1,1)/ct1
       end if
c
c      Calculate cell potential from dif of solid phase potentials:
       v=xt(6,nj,kk)-xt(6,1,kk)
c
c      Calculate utilization of two electrodes based on coulombs passed:
       if(li.eq.1) then
c      Calculate energy density by running sum of currentxvoltage:
       ed=ed+(Vold+v)*(ts(kk)-ts(kk-1))*cur/2.0d0
       Vold=v
       ut=cur*(ts(kk)-ts(kk-1))/fc/(1.0d0-ep3-epf3-epp3)/n3/h3/ct3+ut
       if (n1 .gt. 0)   !***Need to fix how utilization is calculated for a foil anode
      &ut2=ut2-cur*rr/fc/(1.0d0-ep1-epf1-epp1)/n1/h1/(ct1)
       th=ts(kk)/6.0d01
c
       if(lht.ne.2) call temperature(kk,v,Uoc,Soc)
       tprint=t-273.15
       if(lpow.ne.0) then
c      ! isothermal peak power output:
       write (2,309) v,ca,cur,v*cur
       else
       if (lht.eq.0) then ! T varies, uses htc:
       write (2,309) ut,v,tprint,th,Uoc,cur,qq
      else if (lht.eq.1) then ! calculated htc:
      write (2,309) ut,v,htc,th,Uoc,cur,qq
      else if (lht.eq.2) then ! isothermal output:
c      write (2, 307) ut,v,ca,th,cur,kk,ed
      write (2,307) ut,v,ca,th,cur,ed/tw/3.6d3
      endif
      endif
      endif
      jref=(n1+1+n1+n2)/2
  310 format (1p3e20.6)
      write (3,310) th,xt(6,1,kk)-xt(2,jref,kk)
     %,xt(6,nj,kk)-xt(2,jref,kk)
c
      return
      end
c***********************************************************************
      subroutine sol(nmax,jj)
      implicit real*8(a-h,o-z)
c      This subroutine calculates the solid phase concentration profiles.
      parameter(maxt=900)
      common /calc/ ai(maxt),ai2(maxt),u(223,maxt),ts(maxt),
     1h,h1,h2,h3,hcn,hcp,rr,rrmax
      common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     1ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1
      common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,
     1sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
      dimension cs(50)
c
c      set initial value of solid concentration
      do 88 i=1, 50
c      cs(i)=0.0d0
   88 cs(i)=xt(3,jj,1)
c
c      complete calculations for 50 points along radius of particle
      nmax=nmax-1  ! added
      do 10 i=1,50
      y2=0.02d0*dble(i)
c
      sum1=0.0d0
      do 20 kk=1,nmax
      k=nmax+1-kk
c
      t1=(ts(nmax+1)-ts(k))*dfs1/Rad1/Rad1
      sum2=sum1
c
```

```
c     calculate c bar (r,t1)
      sum1=0.0d0
      r1=1.0d0
c
      do 89 j=1,15
      r1=-r1
      y1=j*j*pi*pi*t1
      y3=j*pi*y2
      if (y1 .gt. 1.50d02) then
      da=0.0d0
      else
      da=expf(-y1)
      end if
   89 sum1=sum1-2.0d0*r1*da*dsin(y3)/j/pi/y2
      sum1=1.0d0-sum1
c
c     perform superposition
c
      cs(i)=cs(i)+(xt(3,jj,k+1)+xt(3,jj,k)-2.0d0*xt(3,jj,1)
     1)*(sum1-sum2)/2.0d0
   20 continue
c
   10 continue
      nmax=nmax+1  ! added
c
      write (2,*) ' '
      write (2,*) 'time is ',ts(nmax)
      write (2,*) ' '
      do 90 i=1, 50, 1
   90 write (2,*) .02*i,' ',cs(i)
c
      return
      end
c*************************************************************************
      subroutine mass(re,rs3,rs1,rf,rpl,rc,rcn,rcp)
      implicit real*8(a-h,o-z)
      parameter(maxt=900)
      common /n/ nx,nt,n1,n2,nj,n3,tmmax
      common /calc/ ai(maxt),ai2(maxt),u(223,maxt),ts(maxt),
     1h,h1,h2,h3,hcn,hcp,rr,rrmax
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     1ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1
      common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
      common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,
     1sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
c
```

```
c     mass of positive electrode
      c1=h3*n3*(re*ep3+rpl*epp3+rs3*(1.0d0-ep3-epf3-epp3)+rf*epf3)
c
c     mass of separator
      s=(re*ep2+rpl*epp2+rc*(1-ep2-epp2))*h2*(n2-1)
c
c     mass of negative electrode
      n1hold = n1
      if (n1 .eq. 0) n1hold = 1
      a1=h1*n1hold*(re*ep1+rpl*epp1+rs1*(1.0d0-ep1-epf1-epp1)+rf*epf1)
c
c     mass of current collectors
      cc1=rcn*hcn+rcp*hcp
c
      tw=c1+s+a1+cc1
c
      return
      end
c********************************************************************
      subroutine temperature(kk,v,Uoc, Soc)
      implicit real*8(a-h,o-z)
      parameter(maxt=900)
      common /n/ nx,nt,n1,n2,nj,n3,tmmax
      common /calc/ ai(maxt),ai2(maxt),u(223,maxt),ts(maxt),
     1h,h1,h2,h3,hcn,hcp,rr,rrmax
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     1ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1
      common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
      common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,
     1sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
      common/temp/ thk,htc,dudt,Cp,dens,tam,g0,ncell,lht,qq
c
c
c     Revised by Karen Thomas August 5, 1999 to calculated the
c     enthalpy potential as an average weighted by the local
c     reaction rate.
c     The entropy and open circuit potential for each electrode
c     should be given in ekin with respect to a Li reference electrode
c     at the same local electrolyte concentration.
c     Caution in using Uoc: it does not have units of volts
c     until the last line of this subroutine.
c     If heat from side reactions is to be included, add the term
c     reaction rate*enthalpy of reaction inside the summation at
c     each electrode.
c     Heat generation is negative if exothermic.
c     The time stepping used to calculate the new temperature has been
```

```
c     modified, so that the new temperature changes due to heat
c     generated or exchanged at the old temperature.
c     If cur = 0, the Uoc = sum(U*local reaction rate)=0 unless the
c     cell is relaxing from a previous charge or discharge.  If Uoc = 0,
c     then v is the open circuit potential.
c
c     Negative Electrode
c
      call ekin(1,kk,0,0)
      Ua=-g0*fc*area1*h1*xx(5,1)
      Sa =-dudt*fc*area1*h1*xx(5,1)
      if (n1 .gt. 1) then
      Ua = 0.0
      Sa = 0.0
      sum =0.0
      do 868 j = 1, n1+1, 1
      call ekin(j,kk,0,0)
      trap = 1.0 !factor for trapezoidal integration
      if ((j .eq. 1) .or. (j .eq. n1+1)) trap = 0.5
      h = h1
      if (j .eq. n1+1) h = h2
      Ua=Ua-trap*fc*area1*h*xx(5,j)*g0 !negative sign needed for reaction rate
      Sa=Sa-trap*fc*area1*h*xx(5,j)*dudt
 868  continue
      endif
c
c     Positive Electrode
c
      Uc = 0.0
      Sc = 0.0
      do 878 j = n1+n2,nj,1
      call ekin(j,kk,0,0)
      trap = 1.0
      if ((j .eq. n1+n2) .or. (j .eq. nj)) trap = 0.5
      Uc=Uc-trap*fc*area3*h3*xx(5,j)*g0
      Sc=Sc-trap*fc*area3*h3*xx(5,j)*dudt
 878  continue
c
      Uoc = Uc+Ua!add because signs different on reaction rate
      Soc = Sc+Sa
c
c     Per cell heat generation
c
      qq=cur*v - Uoc +t*Soc ! heat is negative if exothermic
c
c     The heat transfer coefficient is for heat transferred out of
```

```
c    one side of the cell; it is defined based on cell area.
c    htcc is a per-cell heat transfer coefficient.
c
     if (lht.eq.0) then !cell temperature changes
     htcc=htc/Ncell
     t=t+(rr/(dens*Cp*thk))*(htcc*(tam-t)-cur*v+Uoc-t*Soc) !note change in time
derivative
     else
c
c    Calculate htc instead of temperature: the heat transfer coefficient
c    required to keep the temperature constant is
c    calculated as a function of time.  The heat transfer coef.
c    is calculated for heat transferred out of one side of the
c    cell stack.  Htcc is defined as a per-cell heat transfer
c    coefficient.
c
       if (t.ne.tam) then
        htc=Ncell*(Uoc-v*cur-t*Soc)/(t-tam)
       else
        htc=0.0
       endif
       htcc=htc/Ncell
     endif
c
     if (dabs(cur) .gt. 0.0) then
       Uoc = Uoc/cur
       Soc = Soc/cur
       endif
     return
      end
c*********************************************************************
     double precision function expf(x)
     implicit real*8 (a-h,o-z)
     expf=0.d0
     if(x.gt.-700.d0) expf=dexp(x)
     return
     end
c*********************************************************************
     subroutine ekin(j,kk,lag,utz)
     implicit real*8(a-h,o-z)
c    This subroutine evaluates the Butler-Volmer equations.
     parameter(maxt=900)
     common /n/ nx,nt,n1,n2,nj,n3,tmmax
     common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
    1ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1
     common/power/ ed,Vold,ranode,rcathde
```

```
      common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
      common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,
     1sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
      common/temp/ thk,htc,dudt,Cp,dens,tam,g0,ncell,lht,qq
      common/mat/ b,d
      common/bnd/ a,c,g,x,y
      dimension b(10,10),d(10,21)
      dimension a(10,10),c(10,221),g(10),x(10,10),y(10,10)
c
c     Calculate average open-circuit potential in either
c     electrode if lag=1, otherwise lag=0
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%
c
c     OPEN-CIRCUIT POTENTIAL FUNCTIONS:
c
c     g0 is the open-circuit potential in terms of the solid
c     concentration, xx(3,j), with respect to a lithium metal
c     electrode
c     g1 is the derivative of the open-circuit potential wrt
c     the solid concentration
c
c
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&
c
c     FOR THE NEGATIVE ELECTRODE
c
      if (j .le. n1+1) then
c
      if (lag.eq.1) xx(3,j)=utz*ct1
c
c  &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c
      go to (51,52,53,54,55,56),nneg
   51 go to 111  ! Li foil
   52 go to 112  ! Carbon (petroleum coke)
   53 go to 113  ! MCMB 2510 Carbon (Bellcore)
   54 go to 114  ! TiS2
   55 go to 115  ! Tungsten oxide (LixWO3 with 0<x<x=1
      a7=-0.492465d0
      a8=1.901110d0
      a9=0.157123d0
      a10=0.04738d0
      a11=0.810239d0
      a12=40.0d0
      a13=0.133875d0
cccc  if(xx(3,j).gt.a6*ct3) write (2,*) '#109 in ekin, j=',j
c      g0=a1+a2*dtanh(-a3*xx(3,j)/ct3+a4)-a5*((a6-xx(3,j)/ct3)**a7-
c     1a8)-a9*expf(-a10*((xx(3,j)/ct3)**8.0d0))+a11
c     1*expf(-a12*(xx(3,j)/ct3-a13))
      g0=a1+a2*dtanh(-a3*xx(3,j)/ct3+a4)
     1-a9*expf(-a10*((xx(3,j)/ct3)**8.0d0))+a11
     1*expf(-a12*(xx(3,j)/ct3-a13))+a5*a8
      if(xx(3,j).lt.a6*ct3) g0=g0-a5*((a6-xx(3,j)/ct3)**a7)

c      g1=(1.0d0/ct3)*(-a2*a3/dcosh(-a3*xx(3,j)/ct3+a4)/dcosh(-a3
c     1*xx(3,j)/ct3+a4)+a5*a7*(a6-xx(3,j)/ct3)**(-1.0d0+a7)+
c     1a9*a10*8.0d0*((xx(3,j)/ct3)**7.0d0)*expf(-a10*
c     1(xx(3,j)/ct3)**8.0d0))-a11*a12/ct3*expf(-a12*(xx(3,j)/ct3-a13))
      g1=(1.0d0/ct3)*(-a2*a3/dcosh(-a3*xx(3,j)/ct3+a4)/dcosh(-a3
     1*xx(3,j)/ct3+a4)
     1+a9*a10*8.0d0*((xx(3,j)/ct3)**7.0d0)*expf(-a10*
     1(xx(3,j)/ct3)**8.0d0))-a11*a12/ct3*expf(-a12*(xx(3,j)/ct3-a13))
      if(xx(3,j).lt.a6*ct3)g1=g1+a5*a7*(a6-xx(3,j)/ct3)**(-1.0d0+a7)/ct3

cccc  if(xx(3,j).gt.a6*ct3) write (2, *) 'did it'
c
      if (g0.gt.6.0) then
      g0=6.0d0
      g1=0.0d0
c      write (2,*) 'U theta overflow - positive'
      else if (g0.lt.3.0) then
      g0=3.0d0
      g1=0.0d0
c      write (2,*) 'U theta underflow - positive'
      end if
      go to 98
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
c Nonstoichiometric Vanadium oxide (V6O13)
c based on data from West, Zachau-Christiansen, and Jacobsen,
c Electrochim. Acta vol 28, p. 1829, 1983.
c valid for 0.1 < x < 8.25 in LixV6O13.  Enter csx according to
c LiyVO2.167, where 0.05 < y < 0.95
 211  continue
c
      a1= 3.91007
      a2= 0.04697
      a3= 9.15495
      a4= 5.35279
      a5= 0.06752
      a6= 1.0179
      a7=-0.471987
      a8= 14.7524
      a9= 0.69465
      a10= 0.21481
      a11= 0.0008189
      a12= 22.759
      a13= 1.0 -0.631724
      a14= 26.558
c
      g0=a1+a2*dtanh(-a3*xx(3,j)/ct3+a4)
     1-a9*expf(-a10*((xx(3,j)/ct3)**a14))+a11
     1*expf(-a12*(xx(3,j)/ct3-a13))+a5*a8
      if(xx(3,j).lt.a6*ct3)
     & g0=g0-a5*((a6-xx(3,j)/ct3)**a7)
c
      g1=(1.0d0/ct3)*(-a2*a3)/dcosh(-a3*xx(3,j)/ct3+a4)/
     1 dcosh(-a3*xx(3,j)/ct3+a4)
     1 +a9/ct3*a10*a14*((xx(3,j)/ct3)**(a14-1.0d0))
     1 *expf(-a10*((xx(3,j)/ct3)**a14))
     1 -a11*a12/ct3*expf(-a12*(xx(3,j)/ct3-a13))
      if(xx(3,j).lt.a6*ct3)
     & g1=g1+a5*a7*(a6-xx(3,j)/ct3)**(-1.0d0+a7)/ct3
c
      if (g0.gt.6.0) then
      g0=6.0d0
      g1=0.0d0
c      write (2,*) 'U theta overflow - positive'
      else if (g0.lt.3.0) then
      g0=3.0d0
      g1=0.0d0
c      write (2,*) 'U theta underflow - positive'
      end if
c
      go to 98
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
c
c     KINETIC EXPRESSIONS FOR THE POSITIVE ELECTRODE
c
c     h0 is the exchange current density (A/m2)
c     h1 is the derivative of io wrt solid concentration, xx(3,j)
c     h2 is the derivative of io wrt electrolyte concen., xx(1,j)
c
c &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c
c   NONAQUEOUS LIQUIDS
c
  98 if (lag.eq.1) go to 99
      alpha=0.5d0
      alphc=0.5d0
      h0=rka3*dsqrt(xx(1,j))*dsqrt(ct3-xx(3,j))*dsqrt(xx(3,j))
      h1=-rka3*dsqrt(xx(1,j))*dsqrt(ct3-xx(3,j))*dsqrt(xx(3,j))*(1.0d0/
     1(ct3-xx(3,j))-1.0d0/xx(3,j))/2.0d0
      h2=rka3*dsqrt(ct3-xx(3,j))*dsqrt(xx(3,j))/dsqrt(xx(1,j))/2.0d0
c
c &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c
c   POLYMER
c
c   alpha=0.5d0
c   alphc=0.5d0
c   h0=rka3*dsqrt(xx(1,j))*dsqrt(cmax-xx(1,j))*dsqrt(ct3-xx(3,j))
c  1*dsqrt(xx(3,j))
c   h1=-rka3*dsqrt(xx(1,j))*dsqrt(cmax-xx(1,j))*dsqrt(ct3-xx(3,j))
c  1*dsqrt(xx(3,j))*(1.0d0/(ct3-xx(3,j))-1.0d0/xx(3,j))/2.0d0
c   h2=-rka3*dsqrt(xx(3,j))*dsqrt(ct3-xx(3,j))*dsqrt(cmax-xx(1,j))
c  1*dsqrt(xx(1,j))*(1.0d0/(cmax-xx(1,j))-1.0d0/xx(1,j))/2.0d0
c
c &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c
      end if
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%
      if (lag.eq.1) go to 99
c
      r1=alpha*frt
      if(j.le.n1+1) then
         an1=1.0d0
         an2=0.0d0
        else
         an1=0.0d0
         an2=1.0d0
      endif

      r2=r1*(xx(6,j)-xx(2,j)-g0-fc*xx(5,j)*(an1*ranode+an2*rcathde))
c
      de=-2.d0*r2-r2**3/3.d0
      if(dabs(r2).gt.200.d0) then
      if(r2.gt.200.d0) de=7.d86
      if(r2.lt.-200.d0) de=-7.d86
      pe=7.d86
      else
      if(dabs(r2).gt.1.0d-7) de=expf(-r2)-expf(r2)
      pe=expf(-r2)+expf(r2)
      endif
      b(3,1)=h2*de
      b(3,2)=h0*r1*pe
      b(3,6)=-b(3,2)
      b(3,3)=h1*de+h0*r1*g1*pe
      b(3,5)=1.0d0+fc*b(3,2)*(an1*ranode+an2*rcathde)
      g(3)=-h0*de-xx(5,j)
c
   99 return
      end
c***********************************************************************
      subroutine prop(nj,n2,n1)
      implicit real*8(a-h,o-z)
      parameter(maxt=900)
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     1ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1
      common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
      common/tprop/df(221),cd(221),tm(221),
     1ddf(221),dcd(221),dtm(221),dfu(221),d2fu(221)
      common/temp/ thk,htc,dudt,Cp,dens,tam,g0,ncell,lht,qq
c
      do 99 j=1,nj
      ee=ep2+epp2
      if(j .lt. n1+2 .and. n1 .gt. 0) ee=ep1+epp1
      if(j .gt. n2+n1) ee=ep3+epp3
      go to (1,2,3,4,5,6,7,8,9,10,11),nprop
    1 go to 101  ! AsF6 in methyl acetate
    2 go to 102  ! Perchlorate in PEO
    3 go to 103  ! Sodium Triflate in PEO
    4 go to 104  ! LiPF6 in PC (Sony cell simulation)
    5 go to 105  ! Perchlorate in PC (West's simulation)
    6 go to 106  ! Triflate in PEO
    7 go to 107  ! LiPF6 in EC/DMC (Bellcore)
    8 go to 108  ! LiPF6 in EC/DMC (Bellcore) cell #2
  9    go to 109 !Ideal polymer, t+ = 1.0
 10    go to 110 !LiTFSI in PEMO (from Steve Sloop, 1999)
 11    go to 111 ! LiPF6 in EC:DMC
c
c &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c     AsF6 in methyl acetate
c
c     diffusion coefficient of the salt (m2/s)
```

```
 101 df(j)=(ee**1.5d0)*1.54d-09
      ddf(j)=0.0d0
c     conductivity of the salt (S/m)
      cd(j)=2.5d0*(ee**(1.5d0))
      dcd(j)=0.0d0
c     transference number of lithium
      tm(j)=0.20d0
      dtm(j)=0.0d0
c     activity factor for the salt
      dfu(j)=0.0d0
      d2fu(j)=0.0d0
      go to 99
c  &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c     Perchlorate in PEO
c
c     diffusion coefficient of the salt (m2/s)
 102 df(j)=(ee**1.5d0)*1.78d-12
      ddf(j)=0.0d0
c     conductivity of the salt (S/m)
      cd(j)=1.6d-02*ee**(1.5d0)
      dcd(j)=0.0d0
c     transference number of lithium
      tm(j)=0.10d0
      dtm(j)=0.0d0
c     activity factor for the salt
      dfu(j)=0.0d0
      d2fu(j)=0.0d0
      go to 99
c  &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c     Sodium Triflate in PEO
c
 103 r0=1.3041d-07
      r1=4.4978d-07
      r2=-3.1248d-07
      r3=-2.2383d-07
      r4=8.9264d-09
c
c     diffusion coefficient of the salt (m2/s)
      df(j)=0.0001d0*(ee**1.5d0)*(r0+r1*xx(1,j)/1000+
     1r2*((xx(1,j)/1000)**0.5d0) + r3*((xx(1,j)/1000)**1.5d0)
     1+ r4*((xx(1,j)/1000)**3.0d0))
      ddf(j)=0.0001d0*(ee**1.5d0)*(r1/1000d0 +
     10.5d0*r2*(xx(1,j)**(-0.5d0))/(1000.0d0**0.5d0)
     1+ 1.5d0*r3*(xx(1,j)**0.5d0)/1000d0**1.5d0 +
     13.0d0*r4*(xx(1,j)**2.0d0)/1000d0**3.0d0)
      if (xx(1,j).ge.3.0d03) then
      df(j)=(ee**1.5d0)*1.6477d-12
      ddf(j)=0.0d0
      end if
c      conductivity of the salt (S/m)
      r7=4.32d-05
      r8=0.00017d0
      r9=0.000153d0
      r10=3.73d-05
c
      cd(j)=100*(ee**(1.5d0))*(r7+r8*xx(1,j)/1000+r9*xx(1,j)*xx(1,j)
     1/1000000+r10*xx(1,j)*xx(1,j)*xx(1,j)/1000000000)
      dcd(j)=100*(ee**(1.5d0))*(r8/1000+2.0*r9*xx(1,j)/1000000+3.0*r10
     1*xx(1,j)*xx(1,j)/1000000000)
c
c      transference number of lithium
c
      if(xx(1,j).lt.0.3d03) then
      r5=0.32141d0
      r6=2.5768d0
      r11=71.369d0
      r12=643.63d0
      r13=1983.7d0
      r14=2008.d0
      r15=287.46d0
      tm(j)=r5-r6*xx(1,j)/1000.+r11*xx(1,j)*xx(1,j)/1000000.
     1-r12*((xx(1,j)/1000.)**(3.0d0))+r13*((xx(1,j)/1000.)**4.0d0)
     1-r14*((xx(1,j)/1000.)**(5.0d0))+r15*((xx(1,j)/1000.)**6.0d0)
      dtm(j)=-r6/1000.+2.0d0*r11*xx(1,j)/1000000.-
     13.0d0*r12*(xx(1,j)**2.0d0)/(1000.**3.0d0) +
     14.0d0*r13*(xx(1,j)**3.0d0)/(1000.**4.0d0) -
     15.0d0*r14*(xx(1,j)**4.0d0)/(1000.**5.0d0) +
     16.0d0*r15*(xx(1,j)**5.0d0)/(1000.**6.0d0)
      else
      tm(j)=0.0d0
      dtm(j)=0.0d0
      end if
c
      if(xx(1,j).ge.0.70d03) then
      r5=4.5679d0
      r6=4.506d0
      r11=0.60173d0
      r12=1.0698d0
      tm(j)=-r5+r6*expf(-((xx(1,j)/1000.-r11)/r12)**2.)
      dtm(j)=-r6*(xx(1,j)/1000.-r11)*2.
     1*expf(-((xx(1,j)/1000.-r11)/r12)**2.)/r12/r12/1000.
      end if
```

```
c
      if(xx(1,j).ge.2.58d03) then
      tm(j)=-4.4204d0
      dtm(j)=0.0d0
      end if
c
c     activity factor for the salt:  (dlnf/dc) and (d2lnf/dc2)
c
      if(xx(1,j).gt.0.45d03) then
      r17=0.98249d0
      r18=1.3527d0
      r19=0.71498d0
      r20=0.16715d0
      r21=0.014511d0
      thermf=r17-r18*xx(1,j)/1000.+r19*xx(1,j)*xx(1,j)/1000000.-
     1r20*xx(1,j)*xx(1,j)*xx(1,j)/1000000000.+r21*xx(1,j)*xx(1,j)
     1*xx(1,j)*xx(1,j)/1000000000000.
      dthermf=-r18/1000.+2.*r19*xx(1,j)/1000000.-
     13.*r20*xx(1,j)*xx(1,j)/1000000000.+4.*r21*xx(1,j)*xx(1,j)
     1*xx(1,j)/1000000000000.
      end if
c
      if(xx(1,j).le.0.45d03) then
      r23=0.99161d0
      r24=0.17804d0
      r25=55.653d0
      r26=303.57d0
      r27=590.97d0
      r28=400.21d0
      thermf=r23-r24*xx(1,j)/1000.-r25*xx(1,j)*xx(1,j)/1000000.+
     1r26*xx(1,j)*xx(1,j)*xx(1,j)/1000000000.-r27*xx(1,j)*xx(1,j)
     1*xx(1,j)*xx(1,j)/1000000000000.+r28*xx(1,j)*xx(1,j)*xx(1,j)
     1*xx(1,j)*xx(1,j)/1000000000000000.
      dthermf=-r24/1000.-2.*r25*xx(1,j)/1000000.+
     13.*r26*xx(1,j)*xx(1,j)/1000000000.-4.*r27*xx(1,j)
     1*xx(1,j)*xx(1,j)/1000000000000.+5.*r28*xx(1,j)*xx(1,j)
     1*xx(1,j)*xx(1,j)/1000000000000000.
      end if
c
      dfu(j)=(-1.+2.*thermf)/xx(1,j)
      d2fu(j)=1./xx(1,j)/xx(1,j)-2.*thermf/xx(1,j)/xx(1,j)+
     12.*dthermf/xx(1,j)
c
      if(xx(1,j).ge.3.00d03) then
      dfu(j)=-0.9520/xx(1,j)
      d2fu(j)=0.9520/xx(1,j)/xx(1,j)
      end if
      go to 99
c  &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c     LiPF6 in PC (Sony cell simulation)
c
c     this is actually the diff coeff for perchlorate
c     diffusion coefficient of the salt (m2/s)
  104 df(j)=(ee**1.5d0)*2.58d-10
      ddf(j)=0.0d0
c     conductivity of the salt (S/m)
c     pmax=0.5409
      pmax=0.035d0
      pu=0.857d0
      aa=1.093
      bb=0.04d0
      rho=1.2041d03
      fun=pmax*((1.0d0/rho/pu)**aa)*expf(bb*((xx(1,j)/rho-pu)**2.0)
     1-(aa/pu)*(xx(1,j)/rho-pu))
      fun2=2.0d0*(bb/rho)*(xx(1,j)/rho-pu)-aa/pu/rho
      cd(j)=0.0001+(ee**1.5d0)*((xx(1,j))**aa)*fun
      dcd(j)=(ee**1.5d0)*fun*(aa*(xx(1,j)**(aa-1.0d0))+(xx(1,j)**aa)
     1*fun2)
c     transference number of lithium
      tm(j)=0.20d0
      dtm(j)=0.0d0
c
c     activity factor for the salt (dlnf/dc and d2lnf/dc2)
      dfu(j)=0.0d0
      d2fu(j)=0.0d0
      go to 99
c  &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c     Perchlorate in PC (West's simulation)
c
c     diffusion coefficient of the salt (m2/s)
  105 df(j)=(ee**1.5d0)*2.58d-10
      ddf(j)=0.0d0
c     conductivity of the salt (S/m)
      pmax=0.542d0
      pu=0.6616d0
      aa=0.855d0
      bb=-0.08d0
      rho=1.2041d03
      fun=pmax*((1.0d0/rho/pu)**aa)*expf(bb*((xx(1,j)/rho-pu)**2.0)
     1-(aa/pu)*(xx(1,j)/rho-pu))
      fun2=2.0d0*(bb/rho)*(xx(1,j)/rho-pu)-aa/pu/rho
      cd(j)=0.0001+(ee**1.5d0)*((xx(1,j))**aa)*fun
      dcd(j)=(ee**1.5d0)*fun*(aa*(xx(1,j)**(aa-1.0d0))+(xx(1,j)**aa)
     1*fun2)
c     transference number of lithium
      tm(j)=0.20d0
      dtm(j)=0.0d0
c     activity factor for the salt
      dfu(j)=0.0d0
      d2fu(j)=0.0d0
      go to 99
c  &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c     Triflate in PEO
c
  106 r0=-5.0891863844d-05
      r1=8.38645199394d-07
      r2=-5.19747901855d-10
      r3=8.0832709407d-14
c     diffusion coefficient of the salt (m2/s)
      df(j)=(ee**1.5d0)*7.5d-12
      ddf(j)=0.0d0
c     conductivity of the salt (S/m)
      cd(j)=(ee**(1.5d0))*100.0d0*(r0 + r1*xx(1,j)
     &+r2*xx(1,j)*xx(1,j)+r3*xx(1,j)*xx(1,j)*xx(1,j))
      dcd(j)=(ee**1.5d0)*100.0d0*(r1 + 2.0d0*r2*xx(1,j) +
     &3.0d0*r3*xx(1,j)**2.0d0)
c     transference number of lithium
c     rough conc. dependence of t+ - highly suspect
      tm(j)=0.0107907d0 + 1.48837d-04*xx(1,j)
      dtm(j)=1.48837d-04
c     activity factor for the salt
      dfu(j)=0.0d0
      d2fu(j)=0.0d0
      go to 99
c  &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c     LiPF6 in EC/DMC and p(VdF-HFP) (Bellcore)
c     This is the 1:2 v/v mixture of EC/DMC (eq. 2 of paper)
c     D and t+ given below were fit from discharge curves
c
c     diffusion coefficient of the salt (m2/s)
c 107 df(j)=(ee**1.5d0)*9.00d-11
  107 df(j)=(ee**3.3d0)*7.50d-11
      ddf(j)=0.0d0
c
c     conductivity of the salt (S/m)
c     This is the conductivity of the liquid + salt only (no polymer)
c
c     kappa (c) for EC/DMC 2:1 with LiPF6 at 25 deg C
```

```
c     cd(j)=(ee**1.5d0)*(0.0911+1.9101*xx(1,j)/1000-1.052*xx(1,j)*
c    1xx(1,j)/1000/1000+0.1554*(xx(1,j)/1000)**3.0d0)
c     derivative of kappa (c) for EC/DMC 2:1 at 25 deg C
c     dcd(j)=(ee**1.5d0)*(1.9101/1000-2.0*1.052*xx(1,j)/1000/1000
c    1+3.0*0.1554/1000*(xx(1,j)/1000)**2.0d0)
c
c     kappa (c) for EC/DMC 1:2 w/ LiPF6 at 25 deg C
c     Note Bruggeman exponent should be adjusted to account for
c     polymer phase - this also affects "fac" parameter in Ohm's
c     law equation number 2
      r1=0.00010793d0
      r2=0.0067461d0
      r3=0.0052245d0
      r4=0.0013605d0
      r5=0.00011724d0
      cd(j)=(ee**3.3d0)*(r1+r2*xx(1,j)/1000.d0
c     cd(j)=(ee**1.5d0)*(r1+r2*xx(1,j)/1000.d0
     1-r3*xx(1,j)*xx(1,j)/1000000.d0
     1+r4*(xx(1,j)/1000.d0)**3.0d0-r5*(xx(1,j)/1000.d0)**4.0d0)*100.d0
      dcd(j)=(ee**3.3d0)*(r2/10-r3*2.0d0*xx(1,j)/10000.d0
c     dcd(j)=(ee**1.5d0)*(r2/10-r3*2.0d0*xx(1,j)/10000.d0
     1+3.0d0*r4*xx(1,j)*xx(1,j)/10000000.d0
     1-0.4d0*r5*(xx(1,j)/1000.d0)**3.0d0)
c
c     transference number of lithium
      tm(j)=0.363d0
      dtm(j)=0.0d0
c
c     activity factor for the salt (dlnf/dc and d2lnf/dc2)
      dfu(j)=0.0d0
      d2fu(j)=0.0d0
      go to 99
c &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c     LiPF6 in EC/DMC and p(VdF-HFP) (Bellcore) cell #2
c     This is the 2:1 v/v mixture of EC/DMC (eq. 1 of paper)
c     D and t+ given below were fit from discharge curves
c
c     diffusion coefficient of the salt (m2/s)
c 108 df(j)=(ee**1.5d0)*9.00d-11
  108 df(j)=(ee**3.3d0)*7.50d-11
      ddf(j)=0.0d0
c
c     conductivity of the salt (S/m)
c
c     kappa (c) for EC/DMC 2:1 w/ LiPF6 at 25 deg C
c     Note Bruggeman exponent should be adjusted to account for
```

```
c     polymer phase - this also affects "fac" parameter in Ohm's
c     law equation number 2
      r1=0.00041253d0
      r2=0.005007d0
      r3=0.0047212d0
      r4=0.0015094d0
c     r5=0.0016018d0
      r5=0.00016018d0
      cd(j)=(ee**3.3d0)*(r1+r2*xx(1,j)/1000.d0
c     cd(j)=(ee**1.5d0)*(r1+r2*xx(1,j)/1000.d0
     1-r3*xx(1,j)*xx(1,j)/1000000.d0
     1+r4*(xx(1,j)/1000.d0)**3.0d0-r5*(xx(1,j)/1000.d0)**4.0d0)*100.d0
      dcd(j)=(ee**3.3d0)*(r2/10-r3*2.0d0*xx(1,j)/10000.d0
c     dcd(j)=(ee**1.5d0)*(r2/10-r3*2.0d0*xx(1,j)/10000.d0
     1+3.0d0*r4*xx(1,j)*xx(1,j)/10000000.d0
     1-0.4d0*r5*(xx(1,j)/1000.d0)**3.0d0)
c
c     transference number of lithium
      tm(j)=0.363d0
      dtm(j)=0.0d0
c
c     activity factor for the salt (dlnf/dc and d2lnf/dc2)
      dfu(j)=0.0d0
      d2fu(j)=0.0d0
      go to 99
c  &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c     Ion Exchange Membrane, t+ = 1.0
 109  continue
      df(j) = (ee**1.5)*1.0d-11
      ddf(j) = 0.0d0
      cd(j) = (ee**1.5)*0.01d0
      dcd(j) = 0.0
      tm(j) = 1.0d0
      dtm(j) = 0.0d0
      dfu(j) = 0.0d0
      d2fu(j) = 0.0d0
      go to 99
c
c &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c     LiTFSI in PEMO from Steve Sloop and John Kerr
 110  continue
      df(j) = (ee**1.5)*((-3.0d-17)*xx(1,j) + 6.0d-13)
      ddf(j) = (ee**1.5)*(-3.0d-17)
      cd(j) = (ee**1.5)*expf((-3.0d-07)*xx(1,j)*xx(1,j)
     & + 6.5d-04*xx(1,j) - 1.12)
      dcd(j) = cd(j)*((-6.0d-07)*xx(1,j) + 6.5d-04)
      tm(j) = 1.6*expf(-0.0017*xx(1,j))
      dtm(j) = tm(j)*(-0.0017)
      dfu(j) = 0.0d0
      d2fu(j) = 0.0d0
      go to 99
c
c &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c LiPF6 in EC:DMC.  D is for LiAsF6 in methyl formate,
c corrected for viscosity by Walden's Rule, from the
c Organic Electrolyte Handbook (Janz).  t+ is for LiClO4
c in various electrolytes.  cd is from measurements made at
c Bellcore, as reported in Marc Doyle's dissertation.
 111  continue
      df(j) = (ee**1.5)*1.7d-10
      ddf(j) = 0.0d0
      cd(j) = (ee**1.5)*(0.0911+1.9101*xx(1,j)/1000.0 -
     & 1.052*((xx(1,j)/1000.0)**2.0) +
     & 0.1554*((xx(1,j)/1000.0)**3.0))
      dcd(j) = (ee**1.5)*(1.9101/1000.0 -
     & 2.0*1.052*xx(1,j)/1000.0/1000.0
     & + 0.1554*3.0*((xx(1,j)/1000.0)**2.0)/1000.0)
      tm(j) =0.3d0
      dtm(j) = 0.0d0
      dfu(j) = 0.0d0
      d2fu(j) = 0.0d0
c &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
   99 continue
c
      return
      end
c**********************************************************************
c     That's All Folks!
c**********************************************************************
```

## Appendix B – A sample input file for running a battery simulation

```
50 ! lim, limit on number of iterations
100.d-06 ! h1, thickness of negative electrode (m)
52.d-06 ! h2, thickness of separator (m)
200.d-06 ! h3, thickness of positive electrode (m)
25.d-06 ! hcn, thickness of negative electrode current collector (m)
25.d-06 ! hcp, thickness of positive electrode current collector (m)
80 ! n1, number of nodes in negative electrode.  Set to 0 to flag FOIL mode.
40 ! n2, number of nodes in separator
80 ! n3, number of nodes in positive electrode
298.13 ! T, temperature (K)
2000. ! xi(1,1), initial salt concentration (mol/m3)
0.5635 ! x, initial stoichiometric parameter for negative (ignored if n1=0)
0.1705 ! y, initial stoichiometric parameter for positive
120.0 ! tmmax, maximum time step size (s)
.00 ! vcut, cutoff potential
3.9d-14 ! dfs1, diffusion coefficient in negative solid (m2/s)
1.0d-13 ! dfs3, diffusion coefficient in positive solid (m2/s)
1.25d-05 ! Rad1, radius of negative particles (m) (ignored if n1 = 0)
8.5d-06 ! Rad3, radius of positive particles (m)
0.357 ! ep1, volume fraction of electrolyte in negative electrode (set to 0 if  n1=0)
0.146 ! epp1, volume fraction of polymer in negative electrode (set to 0 if n1 = 0)
0.026 ! epf1, volume fraction of inert filler in negative electrode (set to 0 if n1 = 0)
0.724 ! ep2, volume fraction of electrolyte in separator
0.276 ! epp2, volume fraction of polymer in separator
0.300 ! ep3, volume fraction of electrolyte in positive electrode
0.000 ! epp3, volume fraction of polymer in positive electrode
0.073 ! epf3, volume fraction of inert filler in positive electrode
100. ! sig1, conductivity of negative matrix (S/m)
3.8d-00 ! sig3, conductivity of positive matrix (S/m)
5.0d-09 ! rka1, rate constant for negative reaction
5.0d-09 ! rka3, rate constant for positive reaction
0.110 ! ranode, negative electrode film resistance (ohm-m2)
0.000 ! rcathde, positive electrode film resistance (ohm-m2)
372.0d0 ! cot1, coulombic capacity of negative material (mAh/g)
148.0d0 ! cot3, coulombic capacity of positive material (mAh/g)
1324. ! re, density of electrolyte (kg/m3)
1900. ! rs1, density of negative insertion material (kg/m3)
4140. ! rs3, density of positive insertion material (kg/m3)
1800. ! rf, density of inert filler (kg/m3)
1780. ! rpl, density of polymer material (kg/m3)
2000. ! rc, density of inert separator material (kg/m3)
8930. ! rcn, density of negative current collector (kg/m3)
2700. ! rcp, density of positive current collector (kg/m3)
6.0 ! htc, heat-transfer coefficient at ends of cell stack (W/m2K)
0.0 ! dUdT, temperature coefficient of open-circuit potential (V/K)
```

```
2000.0 ! Cp, heat capacity of system (J/kg-K)
298.0 ! Tam, ambient air temperature (K)
1 ! ncell, number of cells in a cell stack
2 ! lht, 0 uses htc, 1 calcs htc, 2 isothermal
0 ! il1, 1 for long print-out 0 for short print-out
2 ! il2, prints every il2 th node in long print-out
5 ! il3, prints every il3 th time step in long print-out
1 ! lflag, 0 for electrolyte in separator only, 1 for uniform
0 ! lpow 0 for no power peaks, 1 for power peaks
0 ! jsol calculate solid profiles if 1 < jsol < nj
3 ! nneg see below
7 ! nprop see below
9 ! npos see below
1 ! lcurs, number of current changes
0.875d1 12.d0 1 ! test run for new programs


DUAL LITHIUM ION CELL SIMULATION
lines 34 and 35: cot1,cot3
cot1 coulombic capacity of negative electrode (mAh/g) when x=1 in LixC6
cot3 coulombic capacity of positive electrode (mAh/g) when y=1 in LiyCoO2 (332.8),
Li1+yMn2O4(144.50)

line 50: il1
il1 0 gives short print-out no matter if a run converges or not
1 gives long print-out no matter if a run converges or not
The long print-out stops at t(noncovergence).
2 gives short print-out if a run converges but a long
print-out if the run does not converge.

line 59: lcurs, number of current changes

line 60 onward: cu(i), tt(i), mc(i)
cu(i) The ith value of the current (A/m2) or potential (V) of the discharge
tt(i) The ith value of the time (min) or cutoff potential (V) of the discharge
mc(i) The mode of discharge; 0 for potentiostatic, 1 for galvanostatic
for a given time, 2 for galvanostatic to a cutoff potential

nneg:
1 ! Li foil (use with n1 = 0)
2 ! Carbon (petroleum coke)
3 ! MCMB 2510 carbon (Bellcore)
4 ! TiS2
5 ! Tungsten oxide (LixWO3 with 0<x<0.67)
6 ! Lonza KS6 graphite (Bellcore)
```

```
nprop:
1 ! AsF6 in methyl acetate
2 ! Perchlorate in PEO
3 ! Sodium Triflate in PEO
4 ! LiPF6 in PC (Sony cell simulation)
5 ! Perchlorate in PC (West simulation)
6 ! Triflate in PEO
7 ! LiPF6 in EC/DMC and p(VdF-HFP) (Bellcore)
8 ! LiPF6 in EC/DMC and p(VdF-HFP) (Bellcore) cell #2
9 ! Ion exchange membrane, t+ = 1.0
10 ! LiTFSI in PEMO
11 ! Li PF6 in EC:DMC
```

```
npos:
1 ! TiS2
2 ! Spinel Mn2O4 (lower plateau)
3 ! NaCoO2: Sodium cobalt oxide
4 ! Spinel Mn2O4 (upper plateau)
5 ! Tungsten oxide (LixWO3 with 0<x<0.67)
6 ! CoO2 (Cobalt dioxide)
7 ! V2O5 (Vanadium oxide)
8 ! NiO2 (Nickel dioxide)
9 ! Spinel Mn2O4 (Bellcore)
10 ! V6O13 (Vanadium oxide)
11 ! LiAl0.2Mn1.8O4F0.2 spinel from Bellcore
```

## Appendix C – Battery simulation code modified for electrodes with linear porosity gradients

```
c************************************************************************
**
c    POREGRAD.f
c    Linear porosity matches dualfoil.f
c    NO extra fac correction throughout electrode
c    **** dedx term successfully added! ****
c    POROSITY GRADIENT IN EFFECT
c
c
c
c    dual.f  (version 3.0)  January 10, 2000
c    Dual lithium ion insertion cell
c    Includes Bellcore physical properties
c  Copyright Marc Doyle and John Newman 1998.
c  You may make a copy of this program which you may
c  personally and freely use in its unaltered form.
c  You may distribute this program subject to the
c  conditions that it be made freely available and
c  that any duplication of this program must be
c  essentially unaltered and must include this notice.
c
c  We make no warranties, express or implied, that
c  this program is free of errors or that it will
c  meet the requirements of your application.  The
c  author and publisher disclaim all liablility for
c  direct or consequential damages resulting from
c  use of this program.
c
c    Revised June, 1998, to include double-layer capacitance in
c    each electrode and to correct a factor of two in Ohm's law.
c
c  Note: For lflag=0, the model works only for initially zero current.
c
c    Revised Feb. 12, 1999:
c    - if n1 = 0, then code treats the negative electrode as metal foil.
c    - subroutine cellpot does not calculate utilization of foil electrode
c    - Changed read and print statements.
c      To run, simply type "webdual", then enter
c      input and output file names when prompted.
c    - double layer capacitance is not currently calculated at
c    a foil electrode
c**********************************************************************
```

```
      implicit real*8(a-h,o-z)
      character *30 filin, filout
      parameter(maxt=900)
common /n/ nx,nt,n1,n2,nj,n3,tmmax
      common /calc/ ai(maxt),ai2(maxt),u(223,maxt),ts(maxt),
     &    h,h1,h2,h3,hcn,hcp,rr,rrmax
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     &    ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1,ep(221),
     &    epp(221),epf(221)
      common/power/ ed,Vold,ranode,rcathde
      common/ssblock/ xp0(6),xx0(6,221),term(221),fj(221)
      common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
      common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,sig(221),area(221),
     &    sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
      common/tprop/df(221),cd(221),tm(221),
     &    ddf(221),dcd(221),dtm(221),dfu(221),d2fu(221)
      common/temp/ thk,htc,dudt,Cp,dens,tam,g0,ncell,lht,qq
      dimension terms(221),tt(200),cu(200),mc(200),tot(200)
   44 format(/'           mass  = ',f7.4,' kg/m2')
   45 format(' specific energy  = ',f8.2,' W-h/kg')
   46 format(' specific power   = ',f8.2,' W/kg')
c
      open(3,file='halfcells',status='unknown')
      write (3,*) 'time      negative       positive'
      print *, 'Enter input file name, press return'
read *, filin
c     open (1, FILE = 'input.in', status = 'old')
      open (1, FILE = filin, status = 'old')
      print *, 'Enter output file name, press return'
      read *, filout
c     open (2, file = 'output.out', status = 'unknown')
      open (2, file = filout, status = 'unknown')
c
c     n is number of equations
      n=6
      lim2=20

      data fc/96487.0d0/, r/8.314d0/, pi/3.141592653589d0/
      data ed/0/, Vold/0/
c
c*****************************************************************
c     read in parameters and boundary conditions
c
      read (1,*) lim          !limit on number of iterations
      read (1,*) h1           !thickness of negative electrode (m)
      read (1,*) h2           !thickness of separator (m)
```

```
    read (1,*) h3           !thickness of positive electrode (m)
read (1,*) hcn          !thickness of negative electrode current collect
or (m)
    read (1,*) hcp          !thickness of positive electrode current collect
or (m)
    thk=h1+h2+h3
    read (1,*) n1           !number of nodes in negative electrode
c   If negative electrode is metal foil, let n1 = 0
    read (1,*) n2           !number of nodes in separator
    read (1,*) n3           !number of nodes in positive electrode
    read (1,*) t            !temperature (K)
    write (2, 1101) lim,1.d6*h1,1.d6*h2,1.d6*h3,1.d6*hcn,1.d6*hcp
   &    ,n1,n2,n3,t
    n2=n2+1
    nj=n1+n2+n3
c
    read (1,*) xi(1,n1+2)    !initial concentration (mol/m3)
c   guess for PHI2
    xi(2,1)=0.05d0
    xi(2,nj)=0.0d0
    read (1,*) csx          !initial stochiometric parameter for negative
    read (1,*) csy          !initial stochiometric parameter for positive
    read (1,*) tmmax          !maximum time step size (s)
read (1,*) vcut         !cutoff potential
    read (1,*) dfs1         !diffusion coefficient in negative solid (m2/s)
    read (1,*) dfs3         !diffusion coefficient in positive solid (m2/s)
    read (1,*) Rad1          !radius of negative particles (m)
c   If negative electrode is metal foil, let Rad1 = 1.0
    read (1,*) Rad3          !radius of positive particles (m)
    write (2,1102) xi(1,n1+2),csx,csy,tmmax,vcut,dfs1,dfs3,
   &    1.d6*Rad1 ,1.d6*Rad3
c   If negative electrode is metal foil, let ep1=epp1=epf1=0.0
    read (1,*) ep1          !volume fraction of electrolyte in negative elec
trode
    read (1,*) epp1          !volume fraction of polymer phase in negative el
ectrode
    read (1,*) epf1         !volume fraction of inert filler in negative ele
ctrode
    read (1,*) ep2          !volume fraction of electrolyte in separator
    read (1,*) epp2          !volume fraction of polymer phase in separator
    read (1,*) ep3          !average volume fraction of electrolyte in posit
ive electrode
    read (1,*) epp3          !volume fraction of polymer phase in positive el
ectrode
    read (1,*) epf3         !volume fraction of inert filler in positive ele
ctrode
```

```
      read (1,*) amaxep        !volume fraction of electrolyte at electrode/ele
ctrolyte interface
      read (1,*) aminep        !volume fraction of electtolyte at electrode/cur
rent collector interface
      read (1,*) sig1         !conductivity of solid negative matrix (S/m)
      read (1,*) sig3         !conductivity of solid positive matrix (S/m)
c     read (1,*) cmax!maximum concentration in electrolyte     (mol/m3)
      read (1,*) rka1         !reaction rate constant for negative reaction
      read (1,*) rka3         !reaction rate constant for positive reaction
      read (1,*) ranode        !anode film resistance (out of place)
      read (1,*) rcathde       !cathode film resistance (out of place)
c     read (1,*) il4 !1 for polymer, 0 for liquid electrolyte
      read (1,*) cot1         !coulombic capacity of negative material (mAh/g)
      read (1,*) cot3         !coulombic capacity of positive material (mAh/g)
      write (2,1103) ep1,epp1,epf1,ep2,epp2,ep3,epp3,epf3,sig1,
     &     sig3,cot1, cot3,cmax,rka1,rka3,il4
      read (1,*) re           ! density of electrolyte (kg/m3)
      read (1,*) rs1           ! density of negative insertion material (kg/m3)
      read (1,*) rs3           ! density of positive insertion material (kg/m3)
      read (1,*) rf           ! density of inert filler (kg/m3)
      read (1,*) rpl           ! density of polymer phase (kg/m3)
read (1,*) rc           ! density of separator material (kg/m3)
      read (1,*) rcn           ! density of negative current collector (kg/m3)
      read (1,*) rcp           ! density of positive current collector (kg/m3)
      write (2,1104) re,rs1,rs3,rf,rpl,rc,rcn,rcp
      read (1,*) htc           !heat transfer coefficient with external medium
(W/m2K)
      read (1,*) dUdT          !temperature coefficient of EMF (V/K)
      read (1,*) Cp           !heat capacity of cell (J/kgK)
      read (1,*) Tam           !ambient temperature (K)
      read (1,*) ncell         !number of cells in a cell stack
      read (1,*) lht          !0 uses htc, 1 calcs htc, 2 isothermal
      write (2,1105) ranode,rcathde,htc,dudt,Cp,tam,ncell,lht
      read (1,*) il1          !1 for long print-out 0 for short print-out
      read (1,*) il2          !1/il2 = fraction of nodes in long print-out
      read (1,*) il3          !1/il3 = fraction of time steps in long print-ou
t
      read (1,*) lflag         ! 0 for electrolyte in separator only, 1 for uni
form
      read (1,*) lpow          ! 0 for no power peaks, 1 for power peaks
      read (1,*) jsol         ! calculate solid profiles if 1<jsol<nj
      read (1,*) nneg          ! designates negative electrode system
      read (1,*) nprop          ! designates electrolyte system
read (1,*) npos        ! designates positive electrode system
      read (1,*) lcurs        ! number of current changes
      write (2,1106) il1,il2,il3,lflag,lcurs
```

```
c
     read (1,*) (cu(i),tt(i),mc(i),i=1,lcurs)
c     cu(i)     operating current density (A/m2)
c     tt(i)     time (min)
c
     print *, 'Now running DUAL...'
 1101 format (i7,' lim, limit on number of iterations'
     &    /1x,f6.2,' h1,  thickness of negative electrode (microns)'
     &    /1x,f6.2,' h2,  thickness of separator (microns)'
     &    /1x,f6.2,' h3,  thickness of positive electrode (microns)'
     &    /1x,f6.2,' hcn, ',
     &    'thickness of negative electrode current collector (microns)'
     &    /1x,f6.2,' hcp, thickness of positive electrode current'
     &    ,' collector (microns)'
     &    /i7,' n1,  number of nodes in negative electrode'
     &    /i7,' n2,  number of nodes in separator'
     &    /i7,' n3,  number of nodes in positive electrode'
     &    /1x,f6.2, ' T,   temperature (K)')
 1102 format (/1x,f6.1,' xi(1,n1+2), initial concentration (mol/m3)'
&/1x,f6.4,' csx,   initial stoichiometric parameter for negative'
     &/1x,f6.4,' csy,   initial stoichiometric parameter for positive'
     &/1x,f6.1,' tmmax, maximum time step size (s)'
     &/1x,f6.2,' vcut,  cutoff potential'
     &/1x,e6.1,' dfs1, diffusion coefficient in negative solid (m2/s)'
     &/1x,e6.1,' dfs3, diffusion coefficient in positive solid (m2/s)'
     &/1x,f10.2,' Rad1,  radius of negative particles (microns)'
     &/1x,f6.2,' Rad3,  radius of positive particles (microns)')
 1103 format (/1x,f6.3,' ep1,'
     &    ,'  volume fraction of electrolyte in negative electrode'
     &    /1x,f6.3,' epp1,'
     &    ,' volume fraction of polymer phase in negative electrode'
     &    /1x,f6.3,' epf1,'
     &    ,' volume fraction of inert filler in negative electrode'
     &/1x,f6.3,' ep2,   volume fraction of electrolyte in separator'
     &/1x,f6.3,' epp2,  volume fraction of polymer phase in separator'
     &    /1x,f6.3,' ep3,'
     &,'  volume fraction of electrolyte in positive electrode'
     &/1x,f6.3,' epp3,'
     &,' volume fraction of polymer phase in positive electrode'
     &/1x,f6.3,' epf3,'
     &,' volume fraction of inert filler in positive electrode'
&/1x,f7.2,' sig1,  conductivity of negative matrix (S/m)'
     &/1x,f7.2,' sig3,  conductivity of positive matrix (S/m)'
     &/1x,f6.2,' cot1,  coulombic capacity of negative material'
     &,' (mAh/g)'
     &/1x,f10.2,' cot3,  coulombic capacity of positive material'
```

```
    &,' (mAh/g)'
    &/1x,f6.0,'  cmax,  maximum concentration in electrolyte (mol/m3)'
    &/1x,e6.1,'  rka1,  reaction rate constant for negative reaction '
    &/1x,e6.1,'  rka3,  reaction rate constant for positive reaction '
    &/i7,'  il4,  1 for polymer, 0 for liquid electrolyte')
 1104 format (/1x,f6.1,'  re,   density of electrolyte (kg/m3)'
    &/1x,f6.1,'  rs1,  density of negative insertion material (kg/m3)'
    &/1x,f6.1,'  rs3,  density of positive insertion material (kg/m3)'
    &/1x,f6.1,'  rf,   density of inert filler (kg/m3)'
    &/1x,f6.1,'  rpl,  density of polymer phase (kg/m3)'
    &/1x,f6.1,'  rc,   density of separator material (kg/m3)'
    &/1x,f6.1,'  rcn,  density of negative current collector (kg/m3)'
    &/1x,f6.1,'  rcp,  density of positive current collector (kg/m3)')
 1105 format (/1x,f6.3,'  ranode,   anode film resistance (ohm-m2)'
    &/1x,f6.3,'  rcathde,  cathode film resistance (ohm-m2)'
    &/1x,f6.2,'  htc,   heat transfer coefficient with'
    &,' external medium (W/m2K)'
&/1x,f6.2,'  dUdT,  temperature coefficient of EMF (V/K)'
    &/1x,f6.1,'  Cp,    heat capacity of cell (J/kg-K)'
    &/1x,f6.2,'  Tam,   ambient temperature (K)'
    &/i7,'  ncell, number of cells in a cell stack'
    &/i7,'  lht,   0 uses htc,  1 calcs htc,  2 isothermal')
 1106 format (/i7,'  il1,   1 for long print-out  0 for short print-out'
    &/i7,'  il2,   prints every il2 th node in long print-out'
    &/i7,'  il3,   prints every il3 th time step in long print-out'
    &/i7,'  lflag, 0 for electrolyte in separator only, 1 for uniform'
    &/i7,'  lcurs, number of current changes')
     write (2,*) ' '
     go to (131,132,133,134,135,136),nneg
  131 write (2,*) 'Li foil'
     go to 137
  132 write (2,*) 'Carbon (petroleum coke)'
     go to 137
  133 write (2,*) 'MCMB 2510 Carbon (Bellcore)'
     go to 137
  134 write (2,*) 'TiS2'
     go to 137
  135 write (2,*) 'Tungsten oxide (LixWO3 with 0<x<0.67)'
     go to 137
136 write (2,*) 'Lonza KS6 graphite (Bellcore)'
  137 go to (101,102,103,104,105,106,107,108,109,110,111),nprop
  101 write (2,*) 'AsF6 in methyl acetate'
     go to 200
  102 write (2,*) 'Perchlorate in PEO'
     go to 200
  103 write (2,*) 'Sodium Triflate in PEO'
```

```
    go to 200
  104 write (2,*) 'LiPF6 in PC (Sony cell simulation)'
    go to 200
  105 write (2,*) 'Perchlorate in PC (West simulation)'
    go to 200
  106 write (2,*) 'Triflate in PEO'
    go to 200
  107 write (2,*) 'LiPF6 in EC/DMC and p(VdF-HFP)'
    go to 200
  108 write (2,*) 'LiPF6 in EC/DMC and p(VdF-HFP) (Bellcore) cell #2'
    go to 200
  109 write (2,*) 'Ion exchange membrane, t+=1.0'
    go to 200
  110 write (2,*) 'LiTFSI in PEMO'
    go to 200
 111 write (2,*) 'LiPF6 in EC:DMC'
    go to 200
  200 go to (201,202,203,204,205,206,207,208,209,210,211),npos
  201 write (2,*) 'TiS2'
    go to 300
  202 write (2,*) 'Spinel Mn2O4 (lower plateau)'
    go to 300
  203 write (2,*) 'NaCoO2:  Sodium Cobalt Oxide'
    go to 300
  204 write (2,*) 'Spinel Mn2O4 (upper plateau)'
    go to 300
  205 write (2,*) 'Tungsten oxide (LixWO3 with 0<x<0.67)'
    go to 300
  206 write (2,*) 'CoO2 (Cobalt dioxide)'
    go to 300
  207 write (2,*) 'V2O5 (Vanadium oxide)'
    go to 300
  208 write (2,*) 'NiO2 (Nickel dioxide)'
    go to 300
  209 write (2,*) 'Spinel Mn2O4 (Bellcore)'
    go to 300
  210 write (2,*) 'V6O13 (Vanadium oxide)'
go to 300
  211 write (2,*) 'LiAl0.2Mn1.8O4F0.2 spinel from Bellcore'
    go to 300
  300 continue
c
c    Fill out porosity gradient array
c
    do 30 i=1,nj
      if (n1 .eq. 0) then
```

```
          ep(1)=ep2
          epp(1)=epp2
          epf(1)=epf2
          goto 486
        endif
        if (i .lt. n1+2) then
          ep(i)=ep1
          epp(i)=epp1
          epf(i)=epf1
        endif
 486      if (i .le. n1+n2 .and. i .ge. n1+2) then
          ep(i)=ep2
          epp(i)=epp2
epf(i)=epf2
        endif
        if (i .gt. n1+n2) then
          ep(i)=amaxep-(amaxep-aminep)*((REAL(i)-(REAL(n1)+
   &          REAL(n2)))/REAL(n3))
          epp(i)=epp3
          epf(i)=epf3
        endif
  30 continue
c    Convert coulombic capacity to total concentrations:
c

      ct1=3.6d03*cot1*rs1/fc
      ct3=3.6d03*cot3*rs3/fc
      xi(3,1)=ct1*csx
      xi(3,n1+1)=xi(3,1)
      xi(3,n2+n1)=ct3*csy
      xi(3,nj)=xi(3,n2+n1)
c
      shape1 = 3.0d0
      shape3 = 3.0d0
      cap1=0.0d0               ! F/m2, capacitance for negative
cap3=0.0d0 ! F/m2, capacitance for positive
c    assume current density linear in electrodes
      cur=cu(1)
      xi(4;1)=0.0d0
      xi(4,n1+1)=cur
      xi(4,n2+n1)=cur
      xi(4,nj)=0.0d0
c
c    Convert times to seconds and sum up times of mode changes
      if (mc(1).lt.2) then
        tot(1)=6.0d01*tt(1)
      else
        tot(1)=0.0d0
      end if
      do 51 i=2,lcurs
        if (mc(i).lt.2) then
          tot(i)=tot(i-1)+6.0d01*tt(i)
        else
          tot(i)=tot(i-1)
        end if
  51  continue
c
c     specific area calculated from geometry
c
c
c     LOOP TO DO AREA AND CONDUCTIVITY AS FUNCTION OF POSITION
c
c

      do 31 ia=1,nj
        if (n1 .eq. 0) then
          area(1)=1.0d0/h1
          sig(1)=sig1
        else
          if (ia .lt. n1+2) then
            area(ia)=shape1*(1.0d0-ep(ia)-epp(ia)-epf(ia))/Rad1
            sig(ia)=sig1*((1.0d0-ep(ia)-epp(ia))**(1.5d0))
          endif
        endif
        if (ia .gt. n1+n2) then
          area(ia)=shape3*(1.0d0-ep(ia)-epp(ia)-epf(ia))/Rad3
          sig(ia)=sig3*((1.0d0-ep(ia)-epp(ia))**(1.5d0))
        endif
        if (ia .eq. n1+n2) then
area(ia)=shape3*(1.0d0-ep(ia+1)-epp(ia+1)-epf(ia+1))
     &          /Rad3
          sig(ia)=sig3*((1.0d0-ep(ia+1)-epp(ia+1))**(1.5d0))
        endif
        if (ia .ge. n1+2 .and. ia .lt.n1+n2) then
          area(ia)=1.1/h1

        endif
c
c     Check to make sure areas are ok
        write (2,*) area(ia)
  31  continue
c
```

```
c
c     DID LOOP TO CALCULATE AREA AND SIGMA(e) AS FUNCTION OF
c     POSITION
c
c
c
c     assume uniform rate of insertion in electrodes
c
      xi(5,1)=cur/fc/h1/area(1)
xi(5,n1+1)=cur/fc/h1/area(n1+1)
      xi(5,n2+n1)=-cur/fc/h3/area(n2+n1)
      xi(5,nj)=-cur/fc/h3/area(nj)
c
c
c     ****OLD DOYLE CONDUCTIVITY FORMULAE****
c
c     sig3=sig3*((1.0d0-ep3-epp3)**(1.5d0))
c     sig1=sig1*((1.0d0-ep1-epp1)**(1.5d0))
c
c
      h2=h2/(n2-1)
      h3=h3/n3
      if (n1 .gt. 1) h1=h1/n1
      h=h2
      frt=fc/(r*t)
c
c     Find initial solid phase potential guesses
c     from initial solid concentrations:
      call ekin(1,kk,1,csx)
      write (2,*) 'open-circuit potential, negative ',g0
      xi(6,1)=g0
xi(6,n1+1)=xi(6,1)
      call ekin(nj,kk,1,csy)
      write (2,*) 'open-circuit potential, positive ',g0
      xi(6,nj)=g0
      xi(6,n2+n1)=xi(6,nj)
c
c     fj is flag to cut off parts of the electrode when c=0
c     Not currently active (10/1/94-CMD)
c     do 52 j=1,nj
c  52 fj(j)=0
c
      write (2,*) ''
      write (2,*) '     DUAL INSERTION CELL VERSION 3.0'
      write (2,*) ''
      if (lht.eq.2) then
      write (2,*) '  util    cell pot    material     time        cur'
     &      ,'     spec energy'
      write (2,*) '    y       (V)       balance      (min)      A/m2'
     &      ,'     W-h/kg'
      else if (lht.eq.0) then
        write (2,*) ' util  ','   cell pot  ',' temp  ','     time  ',
     &       '    U ocp     cur'
write (2,*) '   y  ','     (V)     ',' (C)  ','     (min)  ',
     &       '   (V)  '
      else if (lht.eq.1) then
        write (2,*) ' util  ','   cell pot ','  htcoeff ','   time  ',
     &       '    U ocp     cur         Heat Generation'
        write (2,*) '   y   ','    (V)    ',' (W/m2K) ','   (min) ',
     &       '    (V)       A/m2         W/m2          '
      end if
      write (2,*) ' '
c
c*****************************************************************
c
c     initialize time counting variables
      k=1
      time=0.0d0
      time2=0.0d0
      rr=0.0d0
      ts(1)=0.0d0
c
c     Must activate lpow=1 in data file if you want peak powers:
      kkflag=0
      nflag=0
call guess(lflag)
c     calculate mass (kg/m2) of the cell
      call mass(re,rs3,rs1,rf,rpl,rc,rcn,rcp)
      dens=tw/thk
c
      call comp(n,lim,k,rr,0,nflag,0,jcount)
      call cellpot(k,vv,1,0,lflag)
c
c     rr is the size of a time step.
      if(cap1.eq.0.d0 .and. cap3.eq.0.d0) then
        rr=1.0d0              !  initial time step is 1 second
      else
        rr=1.5d-13            !  initial time step is 1.5d-13 second
      endif
c
      iflag=0
      L=0
```

```
   53 L=L+1
c     do 53 l=1,lcurs
  123 k=k+1
      nt=k-1
c
c     adjust time step to match time of change in current
      time=ts(k-1)+rr
      if(time .ge. tot(l) .and. mc(l).lt.2) then
        rr=tot(l)-ts(k-1)
        time=tot(l)
        iflag=1
      end if
c
  129 ts(k)=ts(k-1)+rr
      call calca(k)
c
      if(mc(l) .gt. 0) then
c     mc is 1 or 2 so run galvanostatically
        dtnow=rr
        call comp(n,lim,k,rr,0,nflag,0,jcount)
        if(rr.lt.0.9999*dtnow) iflag=0
        if(mc(l).lt.2) call cellpot(k,vv,1,0,lflag)
        if(mc(l).eq.2) call cellpot(k,vv,0,0,lflag)
        frt=fc/r/t
c
      else
c     mc is 0 so run potentiostatically
jc2=0
        curlow=0.d0
        curhigh=0.d0
        do 610 j=1,nj
  610   terms(j)=term(j)
        jsig=0
c
  609   jc2=jc2+1
        if(jsig.eq.1) then
          cur=cur+20.d0
        elseif(jsig.eq.2) then
          cur=cur-20.d0
        elseif(jsig.eq.3) then
          cur=0.5d0*(curhigh+curlow)
          curtry=curlow+(curhigh-curlow)/(vhigh-vlow)*(cu(l)-vlow)
          if(curtry.gt.curhigh .and. curtry.lt.curlow) cur=curtry
c     write (2,*) 'curtry ',curtry
        endif
        ts(k)=ts(k-1)+rr
```

```
       do 611 j=1,nj
  611   term(j)=terms(j)
       call calca(k)
 call comp(n,lim,k,rr,0,nflag,0,jcount)
       call cellpot(k,vv,0,0,lflag)
       frt=fc/r/t
       if (jsig.ne.4 .and. dabs(vv-cu(l)) .gt. dabs(1.0d-05)) then
         if (jc2 .gt. lim2) then
           write (2,*) 'this run did not converge'
           stop
         else
           if(vv.gt.cu(l)) then
             vhigh=vv
             curhigh=cur
             if(jsig.le.1) then
               jsig=1
             else
               jsig=3
               if(curhigh.gt.curlow) jsig=4
             endif
           else
             vlow=vv
             curlow=cur
             if(jsig.eq.0 .or. jsig.eq.2) then
               jsig=2
else
               jsig=3
               if(curhigh.gt.curlow) jsig=4
             endif
           endif
 312         format (2i4,f10.5,f10.7,2f10.5)
c    write (2, 312) jc2,jsig,cur,vv,curhigh,curlow
           go to 609
         end if
c
      else
        call cellpot(k,vv,1,0,lflag)
        frt=fc/r/t
      end if
c
    end if
c
c    sign=1.d0
c    if(cu(l).lt.0.d0) sign=-1.d0
c    IF(sign*VV.LT.sign*VCUT) GO TO 100
c    check to see if cutoff potential is exceeded if mc is 2
```

```
    if (mc(l).eq.2) then
IF ((VV.LT.TT(L) .AND. CU(L).GT.0.0) .OR.
   &       (VV.GT.TT(L) .AND. CU(L).LT.0.0)) THEN
        if (dabs(vv-tt(l)) .gt. 1.0d-04) then
c    write (2,*) 'not quite right yet ',vv,rr
          rr=rr/2.0d0
          iflag=1
          go to 129
        else
          time2=time2+rr
          call cellpot(k,vv,1,0,lflag)
          frt=fc/r/t
          iflag=1
         end if
       else
         iflag=0
         time2=time2+rr
         call cellpot(k,vv,1,0,lflag)
         frt=fc/r/t
       end if
     end if
c
c     Increasing time steps:
  rrmax=tmmax
     if(k.le.20) rrmax=10.0d0
     if(jcount.lt.5 .and. k.gt.2 .and. rr.lt.rrmax .and.
     &     iflag.eq.0) then
       rr=rr*2.0d0
       write (2,*) 'next time step increased to ', rr,' (s)'
     end if
c
     if(k.GE.maxt-1) then
       write (2,*) 'kmax=',k,' a larger matrix needed for xt'
       go to 100
     endif
     if (k.GE.501) then          ! trim stored solid concentrations
c     should we have been printing long output as we go along?
        do 92 kk=3,401,2
          kput=(KK+1)/2
          ts(kput)=ts(kk)
        do 92 j=1,nj
        do 92 i=1,n
    92   xt(i,j,kput)=xt(i,j,kk)
        do 93 kk=402,K
          ts(kk-200)=ts(kk)
   do 93 j=1,nj
```

```
        do 93 i=1,n
   93     xt(i,j,kk-200)=xt(i,j,kk)
        k=k-200
      endif
c
      if (iflag .eq. 0) go to 123
      iflag=0
      if (mc(l).eq.2) then
        do 124 m=l,lcurs
  124     tot(m)=tot(m)+time2
        time2=0.0d0
      end if
      IF(L.EQ.LCURS .AND. LCURS.GE.10) THEN
        L=0
        tot(1)=TOT(LCURS)
        if (mc(1).lt.2) TOT(1)=TOT(1)+60.0D0*TT(1)
        do 403 i=2,lcurs
          if (mc(i).lt.2) then
            tot(i)=tot(i-1)+6.0d01*tt(i)
          else
            tot(i)=tot(i-1)
end if
  403     continue
      ENDIF
      if(mc(l+1) .gt. 0) cur=cu(l+1)
c     calculate zero time solution for change in current
      if(mc(l+1) .gt. 0) then
        k=k+1
        ts(k)=ts(k-1)
        rr=0.0
        call comp(n,lim,k,rr,0,nflag,0,jcount)
        call cellpot(k,vv,1,0,lflag)
      endif
      if(cap1.eq.0.d0 .and. cap3.eq.0.d0) then
        rr=2.0d0            !  initial time step is 2 seconds
      else
        rr=1.5d-13           !  initial time step is 1.5d-13 second
      endif
      IF(L.LT.LCURS) GO TO 53
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
c     Additional features section:
c
c     peak-power subroutine:
      if(lpow.eq.1) then
```

```
        il1=0
        call peak(n,lim,cu(1))
      endif
c
c     Solid-phase concentration profiles at given time and position
      if(jsol.gt.0 .and. jsol.lt.nj) call sol(k,jsol)
c
c     print detailed information if requested
 100  if(il1 .eq. 1) call nucamb(il2,il3)
c
c     calculate average energy and power:
      ed=ed/tw/3.6d03
      pow=3.6d03*ed/ts(nt+1)
      write (2,44) tw
      write (2,45) ed
      write (2,46) pow
c
      end
c
c*********************************************************************
subroutine comp(n,lim,kk,tau,kkflag,nflag,lpow,jcount)
      implicit real*8(a-h,o-z)
      parameter(maxt=900)
      common /n/ nx,nt,n1,n2,nj,n3,tmmax
      common /calc/ ai(maxt),ai2(maxt),u(223,maxt),ts(maxt),
     &    h,h1,h2,h3,hcn,hcp,rr,rrmax
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     &    ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1,ep(221),
     &    epp(221),epf(221)
      common/power/ ed,Vold,ranode,rcathde
      common/ssblock/ xp0(6),xx0(6,221),term(221),fj(221)
      common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
      common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,sig(221),area(221),
     &    sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
      common/tprop/df(221),cd(221),tm(221),
     &    ddf(221),dcd(221),dtm(221),dfu(221),d2fu(221)
      common/temp/ thk,htc,dudt,Cp,dens,tam,g0,ncell,lht,qq
      common/mat/ b,d
      common/bnd/ a,c,g,x,y
      dimension b(10,10),d(10,21),termn(221)
      dimension a(10,10),c(10,221),g(10),x(10,10),y(10,10)
  99 format (1h ,//5x,'this run just did not converge'//)
nx=n
      exbrug=1.5d0
      if(nprop.eq.7 .or. nprop.eq.8) exbrug=3.3d0
c
```

```
 666 kadd=0
      if(rr.eq.0 .and. lpow.eq.1) kadd=1
      do 1 j=1,nj
      do 1 i=1,n
         c(i,j)=xt(i,j,kk-1+kadd)
    1 xx(i,j)=xt(i,j,kk-1+kadd)
c     sets first guess to last time-step values
c
c     initialize variables to begin each iteration (jcount is iteration #)
      jcount=0
      do 4 i=1,n
    4 xp(i)=0.0d0
c
    8 j=0
      jcount=jcount+1
c     calculate physical properties
      call prop(nj,n2,n1)
c
c     initialize x and y for iteration
      do 9 i=1,n
      do 9 k=1,n
         x(i,k)=0.0d0
    9 y(i,k)=0.0d0
c
c     store previous iteration of  (xp in xp0)  &  (xx in xx0)
      do 6 i=1,n
         xp0(i)=xp(i)
         if (n1 .lt. 11) then   !foil electrode has 0 node
            xx0(i,1) = xx(i,1)
         else
            xx0(i,n1-10)=xx(i,n1-10)
         endif
    6 xx0(i,n1+n2+10)=xx(i,n1+n2+10)
c
c     for a given iteration, set up governing equations and bc's
c     start at the left interface and move across polymer
c     initialize a,b,d,g arrays for each node
c
   10 j=j+1
c
  do 11 i=1,n
         g(i)=0.0d0
         xx(i,j)=c(i,j)
      do 11 k=1,n
         a(i,k)=0.0d0
         b(i,k)=0.0d0
   11 d(i,k)=0.0d0
c
      if(rr.le.0.0) then        ! treat as a zero time step
        b(1,1)=1.0d0
        g(1)=xt(1,j,kk-1)-c(1,j) ! fix electrolyte concentration
        b(4,3)=1.0d0
        g(4)=xt(3,j,kk-1)-c(3,j) ! fix solid concentration
        go to 112
      endif
c     ______________________________________________________
c     Equation 1, mass balance.
c
      termn(j)=0.
      fac=1.
      dedx=0.0
      zeff1=0.0
zeff2=0.0
      if (j .gt. n1+n2+1) then
        dedx=((epp(n1+n2+1)+ep(n1+n2+1))-(epp(nj)+ep(nj)))/dble(n3)/h3
        zeff1=0.75*dedx*(df(j-1)/(ep(j-1)+epp(j-1))+df(j)/(ep(j)
     &       +epp(j)))/4.
        if (j .lt. nj) then
          zeff2=0.75*dedx*(df(j+1)/(ep(j+1)+epp(j+1))+df(j)/(ep(j)
     &         +epp(j)))/4.
        endif
      endif
      if(j.eq.n1+2 .and. n1 .gt. 0)
     &    fac=((ep(j)+epp(j))/(ep(j-1)+epp(j-1)))**exbrug
      if(j.eq.n1+n2+1) fac=((ep(j)+epp(j))/(ep(j-1)+epp(j-1)))**exbrug
      if (n1.gt.0 .or. j .gt. 1) epn=ep(j)+epp(j)
      hn=h1
      if(j.gt.n1+1) then
        hn=h2
      endif
      if (j.gt.n1+n2) then
        hn=h3
      endif
      if(j.gt.1) then          ! deal with box to left of point.
termn(j)=-(df(j)+fac*df(j-1))*(c(1,j)-c(1,j-1))/hn/2.
     &       -(1.-0.5*(tm(j)+tm(j-1)))*c(4,j-1)/fc
        a(1,1)=epn*hn*0.125/rr-zeff1
     &       -(df(j)+fac*df(j-1))/hn/4.+fac*ddf(j-1)*
     &       (c(1,j)-c(1,j-1))/hn/4.-dtm(j-1)*c(4,j-1)/4./fc
        b(1,1)=epn*hn*0.375/rr+zeff2
     &       +(df(j)+fac*df(j-1))/hn/4.+ddf(j)*(c(1,j)-c(1,j-1))/hn/4.
     &       -dtm(j)*c(4,j-1)/4./fc
```

```
         a(1,4)=(1.-0.5*(tm(j)+tm(j-1)))/2./fc
         g(1)=-epn*hn*(0.375*(c(1,j)-xt(1,j,kk-1))
     &        +0.125*(c(1,j-1)-xt(1,j-1,kk-1)))/rr
       endif
       fac=1.
       if(j.eq.n1+1) then
         if (n1 .gt. 0) fac=((ep(j+1)+epp(j+1))/(ep(j)+epp(j)))**exbrug
         epn=ep(j+1)+epp(j+1)
         hn=h2
       else if(j.eq.n1+n2) then
         fac=((ep(j+1)+epp(j+1))/(ep(j)+epp(j)))**exbrug
         epn=ep(j+1)+epp(j+1)
         hn=h3
       endif
   if(j.ne.nj) then          ! deal with box to right of point.
c     At the foil anode, only the box to the right should be used
        termn(j)=termn(j)-(fac*df(j)+df(j+1))*(c(1,j)-c(1,j+1))/hn/2.
     &        +(1.-0.5*(tm(j)+tm(j+1)))*c(4,j)/fc
        d(1,1)=epn*hn*0.125/rr
     &        -(fac*df(j)+df(j+1))/hn/4.+ddf(j+1)*(c(1,j)-c(1,j+1))/hn/4.
     &        +dtm(j+1)*c(4,j)/4./fc
        b(1,1)=b(1,1)+epn*hn*0.375/rr
     &        +(fac*df(j)+df(j+1))/hn/4.+fac*ddf(j)*(c(1,j)-c(1,j+1))/hn/4.
     &        +dtm(j)*c(4,j)/4./fc
        b(1,4)=b(1,4)-(1.-0.5*(tm(j)+tm(j+1)))/2./fc
        g(1)=g(1)-epn*hn*(0.375*(c(1,j)-xt(1,j,kk-1))
     &        +0.125*(c(1,j+1)-xt(1,j+1,kk-1)))/rr
      endif
      g(1)=g(1)+(termn(j)+term(j))/2.
c     ________________________________________________________________
c     Equation 4, material balance in solid insertion material.
c
      sum=0.0d0
      if(j.le.n1+1) then
        if (n1 .eq. 0) then
          g(4) = xt(3,j,1) - c(3,j)
b(4,3) = 1.0d0
        else
          if (kk .gt. 2) then
            do 54 i=1, kk-2
  54          if(ts(i+1)-ts(i).ne.0.0)
    &sum=sum + (xt(3,j,i+1) - xt(3,j,i))*ai2(kk-i)/(ts(i+1)-ts(i))
          end if
          b(4,3)=ai2(1)/rr
          b(4,5)=1.0d0/Rad1
          g(4)=ai2(1)*xt(3,j,kk-1)/rr - sum -ai2(1)/rr*c(3,j)-c(5,j)/Rad1
          endif
        else if(j.ge.n1+n2) then
          if (kk .gt. 2) then
            do 95 i=1, kk-2
   95        if(ts(i+1)-ts(i).ne.0.0)
       &sum=sum + (xt(3,j,i+1) - xt(3,j,i))*ai(kk-i)/(ts(i+1)-ts(i))
          end if
          b(4,3)=ai(1)/rr
          b(4,5)=1.0d0/Rad3
          g(4)=ai(1)*xt(3,j,kk-1)/rr - sum -c(3,j)*ai(1)/rr-c(5,j)/Rad3
        else
c      In separator
   b(4,3)=1.0d0
          g(4)=-c(3,j)
        endif
c      ______________________________________________________
c      Equation 2, Ohm's law.
c
  112 if(j.le.n1) then
          h=h1
        else if(j.lt.n1+n2) then
          h=h2
        else
          h=h3
        endif
        fac=1.0
        if(j.eq.n1+1 .and. n1 .gt. 0) then
          fac=((ep(j+1)+epp(j+1))/(ep(j)+epp(j)))**exbrug
        endif
        if(j.eq.n1+n2)
       &     fac=((ep(j+1)+epp(j+1))/(ep(j)+epp(j)))**exbrug
c
        if(j.eq.nj .or. fj(j).ne.0.0) then
          b(2,2)=1.0d0
g(2)=-c(2,j)
          go to 12
        endif
        dcf=(xx(1,j+1)-xx(1,j))/h *2.0d0 !factor of 2 added
        r1=(xx(1,j+1)+xx(1,j))/2.0d0
        r4=xx(4,j)
        p1=(tm(j)+tm(j+1))/2.0d0
        p2=(fac*cd(j)+cd(j+1))/2.0d0
        p4=(dfu(j)+dfu(j+1))/2.0d0
        d(2,1)=(1.0d0-p1)*(1.0d0/r1+p4)/h *2.0d0 !factor of 2 added
        b(2,1)=-d(2,1)+((1.0d0-p1)*(d2fu(j)-1.0d0/r1/r1)*dcf
       &     -(1.0d0/r1+p4)*dcf*dtm(j)+frt*r4*fac*dcd(j)/p2/p2)/2.0d0
```

```
      d(2,1)=d(2,1)+((1.0d0-p1)*(d2fu(j+1)-1.0d0/r1/r1)*dcf
     &   -(1.0d0/r1+p4)*dcf*dtm(j+1)+frt*r4*dcd(j+1)/p2/p2)/2.0d0
      d(2,2)=-frt/h
      b(2,2)=frt/h
      b(2,4)=-frt/p2
      g(2)=-(1.0d0-p1)*(1.0d0/r1+p4)*dcf+frt*(c(2,j+1)-c(2,j))/h
     &   +frt/p2*c(4,j)
c     ________________________________________________________________
c     Equation 3, Butler-Volmer kinetics
c
12 if((j.gt.n1+1 .and. j.lt.n1+n2) .or. fj(j).ne.0.0) then
        b(3,5)=1.0d0
        g(3)=-c(5,j)
      else
        call ekin(j,kk,0,0)
        if(j.eq.n1+n2) Uc=g0
        if(j.eq.1) Ua=g0
      endif
c
c     ________________________________________________________________
c
      if(j.ne.1) go to 13
c     ________________________________________________________________
c
c     specify boundary conditions at left interface (j=1)
c     anode/current collector
c
      h=h1
c
      if (n1 .eq. 0) then
        g(5) = cur - c(4,j)
        b(5,4) = 1.0d0
g(6) = fc*c(5,j) - cur
        b(6,5) = -fc
      else
        if(cap1.eq.0.d0) then
          b(5,4)=-1.0d0/h
          b(5,5)=area(j)*fc/2.0d0
          g(5)=c(4,j)/h-area(j)*fc/2.0d0*c(5,j) ! not order h2
        else
          if (rr .eq. 0) then
            g(5)=c(6,j)-c(2,j)-xt(6,j,kk-1)+xt(2,j,kk-1)
            b(5,6)=-1.d0
            b(5,2)= 1.d0
          else
            b(5,4)=-1.0d0/h
            b(5,5)=area(j)*fc/2.0d0
            b(5,6)=area(j)*cap1/rr
            b(5,2)=-area(j)*cap1/rr
            g(5)=(c(4,j)+xt(4,j,kk-1))/h
     &           -area(j)*fc/2.0d0*(c(5,j)+xt(5,j,kk-1))
     &           -area(j)*cap1*(c(6,j)-c(2,j)-xt(6,j,kk-1)
     &           +xt(2,j,kk-1))/rr
          endif
  endif
c
        b(6,6)=-1.0d0/h
        d(6,6)= 1.0d0/h
        b(6,4)=-1.0d0/sig(j)
        g(6)=-cur/sig(j)+c(4,j)/sig(j)-(c(6,j+1)-c(6,j))/h
c
      endif
c
      call band(j)
      go to 10
c
c     ____________________________________________________________
c     specify governing equations in composite anode [1<j<n1+1]
  13 if (j .ge. (n1+1)) go to 110
c
      if(cap1.eq.0.d0) then
        b(5,4)=-1.0d0/h
        a(5,4)=1.0d0/h
        b(5,5)=area(j)*fc
        g(5)=(c(4,j)-c(4,j-1))/h-area(j)*fc*c(5,j)
      else
if (rr .eq. 0) then
          g(5)=c(6,j)-c(2,j)-xt(6,j,kk-1)+xt(2,j,kk-1)
          b(5,6)=-1.d0
          b(5,2)= 1.d0
        else
          b(5,4)=-1.0d0/h/2.0d0
          a(5,4)=1.0d0/h/2.0d0
          b(5,5)=area(j)*fc/2.0d0
          b(5,6)=area(j)*cap1/rr
          b(5,2)=-area(j)*cap1/rr
          g(5)=(c(4,j)-c(4,j-1)+xt(4,j,kk-1)-xt(4,j-1,kk-1))/h/2.0d0
     &         -area(j)*fc/2.0d0*(c(5,j)+xt(5,j,kk-1))
     &         -area(j)*cap1*(c(6,j)-c(2,j)-xt(6,j,kk-1)
     &         +xt(2,j,kk-1))/rr
        endif
      endif
```

```
c
      b(6,6)=-1.0d0/h
      d(6,6)= 1.0d0/h
      b(6,4)=-1.0d0/sig(j)
      g(6)=-cur/sig(j)+c(4,j)/sig(j)-(c(6,j+1)-c(6,j))/h
c
c      do 502 i=3,3
      call band(j)
      go to 10
c     __________________________________________________________
  110 if (j .ne. (n1+1)) go to 120
      if (n1 .eq. 0) go to 120
c
c      Now for the boundary between anode and separator(j=n1+1)
c
      if(cap1.eq.0.d0) then
        b(5,4)=-1.0d0/h
        a(5,4)=1.0d0/h
        b(5,5)=area(j)*fc/2.0d0
        g(5)=(c(4,j)-c(4,j-1))/h-area(j)*fc/2.0d0*c(5,j) ! not order h2
      else
        if (rr .eq. 0) then
          g(5)=c(6,j)-c(2,j)-xt(6,j,kk-1)+xt(2,j,kk-1)
          b(5,6)=-1.d0
          b(5,2)= 1.d0
        else
          b(5,4)=-1.0d0/h/2.0d0
          a(5,4)=1.0d0/h/2.0d0
b(5,5)=area(j)*fc/4.0d0
          b(5,6)=area(j)*cap1/rr*0.5d0
          b(5,2)=-area(j)*cap1/rr*0.5d0
          g(5)=(c(4,j)-c(4,j-1)+xt(4,j,kk-1)-xt(4,j-1,kk-1))/h/2.0d0
     &          -area(j)*fc/4.0d0*(c(5,j)+xt(5,j,kk-1))
     &          -area(j)*cap1*(c(6,j)-c(2,j)-xt(6,j,kk-1)
     &          +xt(2,j,kk-1))/rr*0.5d0
        endif
      endif
c
      b(6,4)=1.0d0
      g(6)=cur-c(4,j)
c
c      do 503 i=3,3
      call band(j)
      go to 10
c     __________________________________________________________
  120 if(j .ge. (n1+n2)) go to 130
```

```
c     specify governing equations [ n1 < j < n2+n1 ]
c     in separator
c
      h=h2
c
      b(5,4)=1.0d0
      g(5)=cur-c(4,j)
c
      b(6,6)=1.0d0
      g(6)=-c(6,j)
c
c     do 504 i=3,3
      call band(j)
      go to 10
c     ________________________________________________
 130 if (j .ne. (n2+n1)) go to 140
c     Boundary between positive and separator(j=n2+n1):
c
      if(cap3.eq.0.d0) then
        b(5,4)=-1.0d0/h
        a(5,4)=1.0d0/h
        b(5,5)=area(j)*fc/2.0d0
        g(5)=(c(4,j)-c(4,j-1))/h-area(j)*fc/2.0d0*c(5,j) ! not order h2
      else
        if (rr .eq. 0) then
          g(5)=c(6,j)-c(2,j)-xt(6,j,kk-1)+xt(2,j,kk-1)
b(5,6)=-1.d0
          b(5,2)= 1.d0
        else
          b(5,4)=-1.0d0/h/2.0d0
          a(5,4)=1.0d0/h/2.0d0
          b(5,5)=area(j)*fc/4.0d0
          b(5,6)=area(j)*cap3/rr*0.5d0
          b(5,2)=-area(j)*cap3/rr*0.5d0
          g(5)=(c(4,j)-c(4,j-1)+xt(4,j,kk-1)-xt(4,j-1,kk-1))/h/2.0d0
     &         -area(j)*fc/4.0d0*(c(5,j)+xt(5,j,kk-1))
     &         -area(j)*cap3*(c(6,j)-c(2,j)-xt(6,j,kk-1)
     &         +xt(2,j,kk-1))/rr*0.5d0
        endif
      endif
c
      d(6,6)=1.0d0/h3
      b(6,6)=-1.0d0/h3
      b(6,4)=-1.0d0/sig(j)
      g(6)=-cur/sig(j)+c(4,j)/sig(j)-(c(6,j+1)-c(6,j))/h3
c
```

```
c    do 505 i=3,3
    call band(j)
go to 10
c    ____________________________________________________________
 140 if (j .eq. nj) go to 16
c
c    specify governing equations [ n2+n1 < j < nj ]
c    composite cathode
c
    h=h3
c
    if(cap3.eq.0.d0) then
      b(5,4)=-1.0d0/h
      a(5,4)=1.0d0/h
      b(5,5)=area(j)*fc
      g(5)=(c(4,j)-c(4,j-1))/h-area(j)*fc*c(5,j)
    else
      if (rr .eq. 0) then
        g(5)=c(6,j)-c(2,j)-xt(6,j,kk-1)+xt(2,j,kk-1)
        b(5,6)=-1.d0
        b(5,2)= 1.d0
      else
        b(5,4)=-1.0d0/h/2.0d0
        a(5,4)=1.0d0/h/2.0d0
b(5,5)=area(j)*fc/2.0d0
        b(5,6)=area(j)*cap3/rr
        b(5,2)=-area(j)*cap3/rr
        g(5)=(c(4,j)-c(4,j-1)+xt(4,j,kk-1)-xt(4,j-1,kk-1))/h/2.0d0
    &        -area(j)*fc/2.0d0*(c(5,j)+xt(5,j,kk-1))
    &        -area(j)*cap3*(c(6,j)-c(2,j)-xt(6,j,kk-1)
    &        +xt(2,j,kk-1))/rr
      endif
    endif
c
    d(6,6)=1.0d0/h
    b(6,6)=-1.0d0/h
    b(6,4)=-1.0d0/sig(j)
    g(6)=-cur/sig(j)+c(4,j)/sig(j)-(c(6,j+1)-c(6,j))/h3
c
c    do 506 i=3,3
    call band(j)
    go to 10
c
c    ____________________________________________________________
c
  16 continue
```

```
c     specify boundary conditions at right interface(j=nj)
c
      if(cap3.eq.0.d0) then
        b(5,4)=-1.0d0/h
        a(5,4)=1.0d0/h
        b(5,5)=area(j)*fc/2.0d0
        g(5)=(c(4,j)-c(4,j-1))/h-area(j)*fc/2.0d0*c(5,j) ! not order h2
      else
        if (rr .eq. 0) then
          g(5)=c(6,j)-c(2,j)-xt(6,j,kk-1)+xt(2,j,kk-1)
          b(5,6)=-1.d0
          b(5,2)= 1.d0
        else
          b(5,4)=-1.0d0/h/2.0d0
          a(5,4)=1.0d0/h/2.0d0
          b(5,5)=area(j)*fc/4.0d0
          b(5,6)=area(j)*cap3/rr*0.5d0
          b(5,2)=-area(j)*cap3/rr*0.5d0
          g(5)=(c(4,j)-c(4,j-1)+xt(4,j,kk-1)-xt(4,j-1,kk-1))/h/2.0d0
     &           -area(j)*fc/4.0d0*(c(5,j)+xt(5,j,kk-1))
     &           -area(j)*cap3*(c(6,j)-c(2,j)-xt(6,j,kk-1)
     &           +xt(2,j,kk-1))/rr*0.5d0
endif
      endif
c
      b(6,4)=1.0d0
      g(6)=-c(4,j)             !  i2 is no longer used at j=nj
c
c     do 507 i=3,3
      call band(j)
      do 607 jj=1,nj
      do 607 i=1,n
  607 c(i,jj)=xx(i,jj)+c(i,jj)
c     ________________________________________________
c
c     check for convergence
c
      do 56 i=1,n
   56 xp(i)=(4.0d0*c(i,2)-3.0d0*c(i,1)-c(i,3))/2.0d0/h1
c
      nerr=0
      do 25 j=1,nj
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
c     shoe horns:
      if(c(1,j).lt.xx(1,j)/100.) c(1,j)=xx(1,j)/100.
      if (c(2,j).lt.(xx(2,j)-0.02)) c(2,j)=xx(2,j)-0.02
      if (c(2,j).gt.(xx(2,j)+0.02)) c(2,j)=xx(2,j)+0.02
      if (c(6,j).lt.(xx(6,j)-0.02)) c(6,j)=xx(6,j)-0.02
      if (c(6,j).gt.(xx(6,j)+0.02)) c(6,j)=xx(6,j)+0.02
      if (c(2,j).gt. 9.9) c(2,j)= 9.9
      if (c(2,j).lt.-9.9) c(2,j)=-9.9
      if (c(6,j).gt. 9.9) c(6,j)= 9.9
      if (c(6,j).lt.-9.9) c(6,j)=-9.9
c
      if (j .ge. n1+n2) then
        if(c(3,j).lt.xx(3,j)/100.) nerr=nerr+1
        if(c(3,j).lt.xx(3,j)/100.) c(3,j)=xx(3,j)/100. ! use cs min
        if(ct3-c(3,j).le.(ct3-xx(3,j))/100.) nerr=nerr+1
    if(ct3-c(3,j).le.(ct3-xx(3,j))/100.) c(3,j)=ct3-(ct3-xx(3,j))/100.
        if(c(3,j).ge.ct3) c(3,j)=0.999999*ct3
        if(c(3,j).lt.1.0d-12) c(3,j)=1.0d-12
c
      else if (j .le. n1+1 .and. n1 .gt. 0) then
        if(c(3,j).lt.xx(3,j)/100.) nerr=nerr+1
        if(c(3,j).lt.xx(3,j)/100.) c(3,j)=xx(3,j)/100. ! use cs min
if(ct1-c(3,j).le.(ct1-xx(3,j))/100.) nerr=nerr+1
    if(ct1-c(3,j).le.(ct1-xx(3,j))/100.) c(3,j)=ct1-(ct1-xx(3,j))/100.
        if(c(3,j).ge.ct1) c(3,j)=0.999999*ct1
        if(c(3,j).lt.1.0d-99) c(3,j)=1.0d-99
      endif
c     to avoid underflow or overflow:
      if(c(1,j).lt.1.0d-12) c(1,j)=1.0d-12
      if(c(1,j).lt.1.0d-10) c(5,j)=0.0
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
c
    do 25 i=1,n
  25 xx(i,j)=c(i,j)
c
    if (jcount .gt. 3*lim .and. rr.eq.0.0d0) then
      write (2, 99)
      stop
    endif
c     ________________________________________
c
c     Decreasing time steps:
    if (jcount .gt. lim .and. rr.gt.0.0d0) then
tau=tau/2.0d0
      rr=tau
      ts(kk)=ts(kk-1)+tau
      write (2,*) 'time step reduced to ', tau, ts(kk)
      if (tau.lt.1.0d-4) then
        if (lpow.eq.1) then !peak power activated
          nflag=1
          go to 66
        endif
        nt=nt-1
        tau=(ts(kk-1)-ts(kk-2))/2.d0
        rr=tau
        kback=0
        ed=ed/tw/3.6d03
        pow=3.6d03*ed/ts(nt+1)
        write (2,*) 'mass is ',tw
        write (2,*) 'energy is ',ed
        write (2,*) 'power is ',pow
        write (2,*) kk-1,' this time step did not converge'
        call nucamb(1,5)
        stop
      else
iflag=0
        call calca(kk)
        go to 666
      end if
c
    else
c
      if(nerr.ne.0) go to 8
      do 55 ii=1,n
        errlim=1.d-10
        if(ii.eq.5) errlim=1.d-16 !change to -14 if problems with convergenc
e
        dxp=dabs( xp(ii)-xp0(ii) )
        if (n1 .lt. 11) then
          n1hold = 1
        else
          n1hold = n1-10
        endif
        dxx=dabs( xx(ii,n1hold) - xx0(ii,n1hold) )
        dxx2=dabs( xx(ii,n1+n2+10)-xx0(ii,n1+n2+10) )
    if(dxx .gt. 1.d-9*dabs(xx(ii,n1hold)).and.dxx.gt.errlim) go to 8
        if(dxx2.gt.1.d-9*dabs(xx(ii,n1+n2+10)).and.dxx2.gt.errlim)
&         go to 8
c    if(dxp.gt.1.d-7*dabs(xp(ii)) .and. dxp.gt.errlim) go to 8
  55    continue
c
```

```
c   ________________________________________________
c
      if(lpow.ne.1) write (2,*) jcount,' iterations required'
c
      do 60 ll=1, nj        ! save present time results in xt()
      do 60 lk=1,n
  60   xt(lk,ll,kk)=xx(lk,ll)
c
c   do 57 j=1,nj
c   if(xx(1,j) .lt. 1.0d-03) fj(j)=1
c   57 if(xx(1,j) .gt. 1.0d-01) fj(j)=0
c
c   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      if(rr.ne.0.0d0) then
        do 58 j=1,nj        ! fix to calculate here for zero time step
  58      term(j)=termn(j)
      else
        do 65 j=1,nj
term(j)=0.
          fac=1.
          if(j.eq.n1+2 .and. n1 .gt. 0)
   &           fac=((ep(j)+epp(j))/(ep(j-1)+epp(j-1)))**exbrug
          if(j.eq.n1+n2+1)
   &           fac=((ep(j)+epp(j))/(ep(j-1)+epp(j-1)))**exbrug
          if (n1 .gt.0 .or. j.gt.1) epn=ep(j)+epp(j)
          hn=h1
          if(j.gt.n1+1) then
            hn=h2
          endif
          if (j.gt.n1+n2) then
            hn=h3
          endif
          if(j.gt.1) term(j)=
   &          -(df(j)+fac*df(j-1))*(c(1,j)-c(1,j-1))/hn/2.
   &          -(1.-0.5*(tm(j)+tm(j-1)))*c(4,j-1)/fc
          fac=1.
          if(j.eq.n1+1) then
            if (n1 .gt. 0) fac=((ep(j+1)+epp(j+1))/
   &             (ep(j)+epp(j)))**exbrug
            epn=ep(j+1)+epp(j+1)
hn=h2
          else if(j.eq.n1+n2) then
            fac=((ep(j+1)+epp(j+1))/(ep(j)+epp(j)))**exbrug
            epn=ep(j+1)+epp(j+1)
            hn=h3
          endif
```

```
 65     if(j.lt.nj) term(j)=term(j)
  &           -(fac*df(j)+df(j+1))*(c(1,j)-c(1,j+1))/hn/2.
  &           +(1.-0.5*(tm(j)+tm(j+1)))*c(4,j)/fc
      endif
c
c
    end if
c
 66 continue
    return
    end
c
c*********************************************************************
    subroutine calca(kk)
    implicit real*8(a-h,o-z)
    parameter(maxt=900)
common /n/ nx,nt,n1,n2,nj,n3,tmmax
    common /calc/ ai(maxt),ai2(maxt),u(223,maxt),ts(maxt),
   &    h,h1,h2,h3,hcn,hcp,rr,rrmax
    common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
   &    ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1,ep(221),
   &    epp(221),epf(221)
    common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,sig(221),area(221),
   &    sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
    dimension ar(4,maxt),bz(6)
c
    do 319 l=1,nt
      ai2(l)=0.0d0
 319 ai(l)=0.0d0
c
    do 70 i=1,kk-1
      ar(1,i)=dfs3*(ts(kk)-ts(i))/Rad3/Rad3
      ar(2,i)=dfs3*(ts(kk)-ts(i+1))/Rad3/Rad3
      ar(3,i)=dfs1*(ts(kk)-ts(i))/Rad1/Rad1
      ar(4,i)=dfs1*(ts(kk)-ts(i+1))/Rad1/Rad1
c
      do 69 m=1,2
        t1=ar(m,i)
t2=ar((m+2),i)
c
        a1=0.0d0
        a12=0.0d0
c
        s=1.644934066848d0
c
c    Bessel's function zeros:
```

```
              bz(1)=2.4048255577d0
              bz(2)=5.5200781103d0
              bz(3)=8.6537281103d0
              bz(4)=11.7915344391d0
              bz(5)=14.9309177086d0
c
              if (shape3.gt.2.0d0) then
c       spherical particles:
                if (t1 .gt. 0.06d0) then
c
                  do 59 j=1,5
                    y1=j*j*pi*pi*t1
 59                 if (y1 .le. 1.5d02) a1=a1+(expf(-y1))/j/j
                    a1=2.0d0*(s-a1)/pi/pi
c
                else
c
                  if (t1.LE.0.0d0) then
                    a1=0.0d0
                  else
                    do 60 j=1,3
                      z=j/dsqrt(t1)
                      call erfc(z,e)
                      y2=j*j/t1
                      if(y2 .ge. 1.5d02) then
                        da=-j*dsqrt(pi/t1)*e
                      else
                        da=expf(-y2)-j*dsqrt(pi/t1)*e
                      end if
 60                 a1=a1+da
                    a1=-t1 + 2.0d0*dsqrt(t1/pi)*(1.0d0+2.0d0*a1)
                  end if
c
                end if
              else
c
        if (shape3.lt.2.0d0) then
c       planar particles:
                if(t1 .gt. 0.06d0) then
c
                  do 61 j=1,5
        da=((-1.0d0)**(j))*(1.0d0 - expf(-(2.0d0*j+1.0d0)*
     &  (2.0d0*j+1.0d0)*pi*pi*t1))/(2.0d0*j+1.0d0)/(2.0d0*j+1.0d0)
 61               a1=a1+da
                  a1=4.0d0*a1/pi/pi
c
```

```
              else
c
                do 62 j=1,3
                  z=j/2.0d0/dsqrt(t1)
                  call erfc(z,e)
    da=((-1.0d0)**(j))*(expf(-j*j/4.0d0/t1)-j/2.0d0*dsqrt(pi/t1)*e)
  62             a1=a1+da
                a1=2.0d0*dsqrt(t1/pi)*(1.0d0+2.0d0*a1)
c
              end if
            else
c    cylindrical particles:
 if (t1.gt.0.06d0) then
c
                do 63 j=1,5
                  da=(1.0d0-expf(-bz(j)*bz(j)*t1))/bz(j)/bz(j)
  63             a1=a1+da
                a1=2.0d0*a1
c
              else
c
                a1=2.0d0*dsqrt(t1/pi)-t1/4.0d0-5.0d0*(t1**1.5d0)/96.0d0/
dsqrt(pi)
   &                    -31.0d0*t1*t1/2048.0d0
c
              end if
            end if
          end if
c
          if (n1 .eq. 0) go to 36
c    (skip calculations of Li diffusion in the solid
c    if the negative electrode is metal foil)
c
          if (shape1.gt.2.0d0) then
c    spherical particles:
            if(t2 .gt. 0.06d0) then
c
              do 64 j=1,5
                y2=j*j*pi*pi*t2
  64           if(y2 .le. 1.5d02) a12=a12+(expf(-y2))/j/j
              a12=2.0d0*(s-a12)/pi/pi
c
            else
c
              if (t2.eq.0.0d0) then
                a12=0.0d0
              else
                do 65 j=1,3
                  z=j/dsqrt(t2)
                  call erfc(z,e)
                  y2=j*j/t2
                  if(y2 .gt. 1.5d02) then
                    da=-j*dsqrt(pi/t2)*e
                  else
                    da=expf(-y2)-j*dsqrt(pi/t2)*e
                  end if
65             a12=a12+da
                a12=-t2 + 2.0d0*dsqrt(t2/pi)*(1.0d0+2.0d0*a12)
              end if
            end if
c
          else
            if (shape1.lt.2.0d0) then
c   planar particles:
              if(t2 .gt. 0.06d0) then
c
              do 66 j=1,5
        da=((-1.0d0)**(j))*(1.0d0 - expf(-(2.0d0*j+1.0d0)
   &*(2.0d0*j+1.0d0)*pi*pi*t2))/(2.0d0*j+1.0d0)/(2.0d0*j+1.0)
  66             a12=a12+da
              a12=4.0d0*a12/pi/pi
c
              else
c
              do 67 j=1,3
                z=j/2.0d0/dsqrt(t2)
                call erfc(z,e)
      da=((-1.0d0)**(j))*(expf(-j*j/4.0d0/t2)-j/2.0d0*dsqrt(pi/t2)*e)
67           a12=a12+da
                a12=2.0d0*dsqrt(t2/pi)*(1.0d0+2.0d0*a12)
c
              end if
            else
c   cylindrical particles:
              if (t2.gt.0.06d0) then
c
                do 68 j=1,5
                  da=(1.0d0-expf(-bz(j)*bz(j)*t2))/bz(j)/bz(j)
  68              a12=a12+da
                a12=2.0d0*a12
c
              else
```

```
c
          a12=2.0d0*dsqrt(t2/pi)-t2/4.0d0-5.0d0*(t2**1.5d0)/96.0d0/dsqrt(pi)
     &          -31.0d0*t2*t2/2048.0
c
                end if
              end if
            end if
c
36       continue
c
          ar(m,i)=a1
  69     ar((m+2),i)=a12
c
        ai(kk-i)=ar(1,i)-ar(2,i)
  70 ai2(kk-i)=ar(3,i)-ar(4,i)
c
      return
      end
c***********************************************************************
      subroutine erfc(z,e)
      implicit real*8(a-h,o-z)
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     &    ep1,epf3,epfl,epp1,epp2,epp3,shape3,shape1,ep(221),
     &    epp(221),epf(221)
c
      a1=0.254829592d0
      a2=-0.284496736d0
      a3=1.421413741d0
      a4=-1.453152027d0
      a5=1.061405429d0
if(z .lt. 2.747192d0) then
        t3=1.0d0/(1.0d0+0.3275911d0*z)
        e=(a1*t3+a2*t3*t3+a3*(t3**3.0d0)+a4*(t3**4.0d0)
     &      +a5*(t3**5.0d0))*expf(-z*z)
      else
c
        if(z .gt. 25.0d0) then
          e=0.0d0
        else
c
          sum=0.0d0
          max=z*z + 0.5
          fac=-0.5d0/z/z
          sum=fac
          tl=fac
          n=1
  10      n=n+1
          if(n .gt. max) go to 15
          tn=tl*(2.0d0*n-1.0d0)*fac
          sum=sum + tn
          if(tn .lt. 1.0d-06) go to 15
          tl=tn
go to 10
  15      e=(expf(-z*z))*(1.0d0+sum)/dsqrt(pi)/z
        end if
      end if
c
      return
      end
c*********************************************************************
      subroutine band(j)
      implicit real*8(a-h,o-z)
      common /n/ nx,nt,n1,n2,nj,n3,tmmax
      common/mat/ b,d
      common/bnd/ a,c,g,x,y
      dimension b(10,10),d(10,21)
      dimension a(10,10),c(10,221),g(10),x(10,10),y(10,10)
      dimension e(10,11,221)
  101 format (15h determ=0 at j=,i4)
      n=nx
      if (j-2)  1,6,8
    1 np1= n + 1
      do 2 i=1,n
        d(i,2*n+1)= g(i)
do 2 l=1,n
        lpn= l + n
    2 d(i,lpn)= x(i,l)
      call matinv(n,2*n+1,determ)
      if (determ)  4,3,4
    3 write (2, 101) j
    4 do 5 k=1,n
        e(k,np1,1)= d(k,2*n+1)
      do 5 l=1,n
        e(k,l,1)= - d(k,l)
        lpn= l + n
    5 x(k,l)= - d(k,lpn)
      return
    6 do 7 i=1,n
      do 7 k=1,n
      do 7 l=1,n
    7 d(i,k)= d(i,k) + a(i,l)*x(l,k)
    8 if (j-nj)  11,9,9
```

```
   9 do 10 i=1,n
     do 10 l=1,n
       g(i)= g(i) - y(i,l)*e(l,np1,j-2)
     do 10 m=1,n
10 a(i,l)= a(i,l) + y(i,m)*e(m,l,j-2)
  11 do 12 i=1,n
       d(i,np1)= - g(i)
     do 12 l=1,n
       d(i,np1)= d(i,np1) + a(i,l)*e(l,np1,j-1)
     do 12 k=1,n
  12 b(i,k)= b(i,k) + a(i,l)*e(l,k,j-1)
     call matinv(n,np1,determ)
     if (determ)  14,13,14
  13 write (2, 101) j
  14 do 15 k=1,n
     do 15 m=1,np1
  15 e(k,m,j)= - d(k,m)
     if (j-nj)  20,16,16
  16 do 17 k=1,n
  17 c(k,j)= e(k,np1,j)
     do 18 jj=2,nj
       m= nj - jj + 1
     do 18 k=1,n
       c(k,m)= e(k,np1,m)
     do 18 l=1,n
  18 c(k,m)= c(k,m) + e(k,l,m)*c(l,m+1)
do 19 l=1,n
     do 19 k=1,n
  19 c(k,1)= c(k,1) + x(k,l)*c(l,3)
  20 return
                                              end
c**********************************************************************
     subroutine matinv(n,m,determ)
     implicit real*8(a-h,o-z)
     common/mat/ b,d
     dimension b(10,10),d(10,21)
     dimension id(10)
     determ=1.0
     do 1 i=1,n
   1 id(i)=0
     do 18 nn=1,n
       bmax=1.1
       do 6 i=1,n
         if(id(i).ne.0) go to 6
         bnext=0.0
         btry=0.0
```

```
        do 5 j=1,n
          if(id(j).ne.0) go to 5
if(dabs(b(i,j)).le.bnext) go to 5
          bnext=dabs(b(i,j))
          if(bnext.le.btry) go to 5
          bnext=btry
          btry=dabs(b(i,j))
          jc=j
   5     continue
        if(bnext.ge.bmax*btry) go to 6
        bmax=bnext/btry
        irow=i
        jcol=jc
   6   continue
      if(id(jc).eq.0) go to 8
      determ=0.0
      return
   8   id(jcol)=1
      if(jcol.eq.irow) go to 12
      do 10 j=1,n
        save=b(irow,j)
        b(irow,j)=b(jcol,j)
  10    b(jcol,j)=save
      do 11 k=1,m
save=d(irow,k)
        d(irow,k)=d(jcol,k)
  11    d(jcol,k)=save
  12    f=1.0/b(jcol,jcol)
      do 13 j=1,n
  13    b(jcol,j)=b(jcol,j)*f
      do 14 k=1,m
  14    d(jcol,k)=d(jcol,k)*f
      do 18 i=1,n
        if(i.eq.jcol) go to 18
        f=b(i,jcol)
        do 16 j=1,n
  16      b(i,j)=b(i,j)-f*b(jcol,j)
        do 17 k=1,m
  17      d(i,k)=d(i,k)-f*d(jcol,k)
  18    continue
      return
      end
c*******************************************************************
    subroutine nucamb(il2,il3)
    implicit real*8(a-h,o-z)
    parameter(maxt=900)
```

```
common /n/ nx,nt,n1,n2,nj,n3,tmmax
      common /calc/ ai(maxt),ai2(maxt),u(223,maxt),ts(maxt),
     &    h,h1,h2,h3,hcn,hcp,rr,rrmax
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     &    ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1,ep(221),
     &    epp(221),epf(221)
      common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
      common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,sig(221),area(221),
     &    sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
      common/tprop/df(221),cd(221),tm(221),
     &    ddf(221),dcd(221),dtm(221),dfu(221),d2fu(221)
      dimension zz(221)
  109 format(f6.1,', ',f15.5,', ',f7.4,', 'g10.4,', ',f6.2,', ',g10.4
     &    ,', ',f10.4)
  309 format(f8.5,', ',f8.5)
   44 format(' t = ',1pe18.6,' min')
c
      do 5 i=1,n1+1
        w=i-1
    5 zz(i) = w*h1*1.0d06
      do 71 i=n1+2,n2+n1
        w=i-(n1+1)
71 zz(i)=zz(n1+1)+w*h2*1.0d06
      do 72 i=n1+n2+1,nj
        w=i-(n1+n2)
   72 zz(i)=zz(n2+n1)+w*h3*1.0d06
c
      do 11 l=1,nt+1
        if(ts(l)-ts(l-1).lt.1.d-6) go to 9
        if (l.lt.nt-5 .and. mod(l-1,il3).ne.0) go to 11
    9   write (2,*) ''
        write (2,*) 'distance concen      PHI2    c solid',
     &       '   current     j      PHI1'
        write (2,*) 'microns  (mol/m3)   (V)    x or y ',
     &       '    (A/m2)  (A/m3)      (V)'
        write (2,44) ts(l)/60.0d0
c
        do 10 j=1,nj,il2
          if (j .le. n1+1) then
            csol=ct1
          else
            csol=ct3
          end if
          if(j.le.n1+1) then
curden=area(j)*fc*xt(5,j,l)
          else if(j.ge.n1+n2) then
          curden=area(j)*fc*xt(5,j,l)
        else
          curden=0.0
        endif
   10   write(2,109) zz(j),xt(1,j,l),xt(2,j,l),xt(3,j,l)/csol,xt(4,j,l)
     &          ,curden,xt(6,j,l)
  11  continue
c
      return
      end
c*******************************************************************
      subroutine guess(lflag)
      implicit real*8(a-h,o-z)
      parameter(maxt=900)
      common /n/ nx,nt,n1,n2,nj,n3,tmmax
      common /calc/ ai(maxt),ai2(maxt),u(223,maxt),ts(maxt),
     &    h,h1,h2,h3,hcn,hcp,rr,rrmax
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     &    ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1,ep(221),
     &    epp(221),epf(221)
common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
      common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,sig(221),area(221),
     &    sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
      common/tprop/df(221),cd(221),tm(221),
     &    ddf(221),dcd(221),dtm(221),dfu(221),d2fu(221)
      dimension del(6)
c
      del(2)=cur*h/2.5
      if (n1 .gt. 1) then
        del(3)=cur/(n1)
      else
        del(3) = cur
      endif
      del(4)=cur/(n3)
      del(5)=(xi(5,nj)-xi(5,1))/(nj-1)
c
      Ua=xi(6,1)
      Uc=xi(6,nj)
      do 73 i=1,(n1+1)
        xi(2,i)=xi(2,1)*(nj-i)/nj
        xi(3,i)=xi(3,1)
        xi(4,i)=xi(4,1)+del(3)*(i-1)
 xi(5,i)=xi(5,1)*area(1)/area(i)
  73 xi(6,i)=xi(2,i)+Ua
c
      do 74 i=(n1+2),(n2+n1-1)
```

```
c    xi(2,i)=xi(2,1)-del(2)*(i-n1-2)
       xi(2,i)=xi(2,1)*(nj-i)/nj
       xi(3,i)=0.0d0
       xi(4,i)=cur
       xi(5,i)=0.0d0
  74 xi(6,i)=0.0d0
c
     do 75 i=(n2+n1),nj
       xi(2,i)=xi(2,1)*(nj-i)/nj
       xi(3,i)=xi(3,nj)
       xi(4,i)=xi(4,n2+n1)-del(4)*(i-n1-n2)
       xi(5,i)=xi(5,n2+n1)*area(n2+n1)/area(i)
  75 xi(6,i)=xi(2,i)+Uc
c
     do 15 i=1,nj
       xt(6,i,1)=xi(6,i)
       xt(5,i,1)=xi(5,i)
       xt(4,i,1)=xi(4,i)
xt(3,i,1)=xi(3,i)
  15 xt(2,i,1)=xi(2,i)
c
     do 16 i=1,nj
       xi(1,i)=xi(1,n1+2)
c    Uniform initial concentration if lflag=1
c    Step function initial concentration if lflag=0
       if(lflag.eq.0 .and. (i.le.n1+1 .or. i.ge.n1+n2))
    &       xi(1,i)=1.0d-01
  16 xt(1,i,1)=xi(1,i)
     return
     end
c*********************************************************************
     subroutine peak(n,lim,curr)
     implicit real*8(a-h,o-z)
     parameter(maxt=900)
     common /n/ nx,nt,n1,n2,nj,n3,tmmax
     common /calc/ ai(maxt),ai2(maxt),u(223,maxt),ts(maxt),
    &    h,h1,h2,h3,hcn,hcp,rr,rrmax
     common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
    &    ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1,ep(221),
    &    epp(221),epf(221)
common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
     common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,sig(221),area(221),
    &    sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
 311 format(f8.5,', ',f7.3,',  ',f8.3,', ',f8.5)
c
c    Peak power current ramp section:
      write (2,*) ' '
      write (2,*) '    PEAK POWER '
      write (2,*) ' '
      write (2,*) 'cell pot  ',' current','   power   min pot'
      write (2,*) ' (V)  ','   (A/m2)','  (W/m2)     (V)'
c
c     Duration of current pulse is 30 seconds.
c
      vcut=2.8d0
      curmin=0.d0
      pwrpmax=0.d0
      rrmax=30.0d0
      cur=curr
  127 kcount=0
      fact=20.0d0
      curmax=0.d0
vfmax=0.d0
      k=nt+2
      do 126 j=1,nj
      do 126 i=1,n
  126 xt(i,j,k)=xt(i,j,k-1)
      ppow=0.0
      ii=0
c     Ramp current:
c
 128  continue
      if (ii.gt.60) return
c
      energ=0.0
 130  ii=ii+1
      cur=cur+fact             ! start a new current density
      if(curmax.ne.0.d0) then
        cur=0.5d0*(curpmax+curmax)
        if(mod(ii,2).eq.1) cur=0.5d0*(curpmax+curmin)
        if(cur.eq.0.d0) cur=0.5d0*(curmax+curmin)
        if(vfmax.gt.0.d0 .and. vfmax.lt.vcut) then
          curtry=curpmax+(vcut-vvpmax)/(vfmax-vvpmax)*(curmax-curpmax)
     &         +0.01d0*(curmax-curpmax)*dble(mod(ii,3)-1)
   if(curtry.lt.curmax .and. curtry.gt.curpmax) cur=curtry
        endif
        if(vfmax.gt.vcut .and. curmin.gt.0.0) then
         v2=pwrmax/curmax
         cur2=curmax
         v1=pwrmin/curmin
         cur1=curmin
         vm=pwrpmax/curpmax
```

```
          curm=curpmax
          resis=-((pwrpmax-pwrmin)/(curpmax-curmin)
     &     -(pwrpmax-pwrmax)/(curpmax-curmax))/(curmin-curmax)
          Uop=(pwrpmax-pwrmin)/(curpmax-curmin)+resis*(curpmax+curmin)
          curtry=Uop/2.d0/resis+0.1d0*(curmax-curmin)*dble(mod(ii,3)-1)
          write (2,*) 'curtry= ',curtry,resis,Uop
          if(curtry.lt.curmax .and. curtry.gt.curmin) cur=curtry
        endif
      endif
      write (2,*) ' cur= ',cur,curmin,curpmax,curmax
      kkflag=0
      iflag=0
      nflag=0
      k=nt+2
timpk=0.0d0
      rr=0.0d0
      ts(k)=ts(k-1)
      call comp(n,lim,k,rr,kkflag,nflag,1,jcount)
      call cellpot(k,vv,0,1,lflag)
      vlast=vv
      rr=0.2d0
  129 kkflag=kkflag+1
      k=k+1
      ts(k)=ts(k-1)+rr
      call calca(k)
c
      call comp(n,lim,k,rr,kkflag,nflag,1,jcount)
c
      if (nflag.eq.1.and.kcount.lt.20) then
        write (2,*) 'Peak current decreased',kcount,fact
        if(cur.lt.curpmax) then
          write (2,*) 'Convergence on power failed; already converged at a hig
h
     &         er current'
          write (2,*) 'Best results obtained are:'
          write (2, 311) pwrpmax/curpmax,curpmax,pwrpmax,vvpmax
write (2,*) ' finished with vcut= ',vcut
          return
        endif
        curmax=cur
        vfmax=0.d0
        go to 128
      endif
      if (kcount.ge.10) return
c
      call cellpot(k,vv,0,1,lflag)
      energ=energ+(vlast+vv)*(ts(k)-ts(k-1))*cur/2.0d0
c
      timpk=timpk+rr
      if (dabs(timpk-30.0d0).gt.0.1) then
c
        if (timpk.lt.30.0) then
          vlast=vv
c      Increasing time steps:
          if(jcount.lt.6 .and. kkflag.gt.5 .and. (2.0d0*rr
     &          +timpk).lt.30.0d0 .and. iflag.eq.0) then
            rr=rr*2.0d0
c      write (2,*) 'next time step increased to ', rr,'(s)'
end if
          if(timpk+rr.gt.30.0) iflag=1
          if(timpk+rr.gt.30.0) rr=30.0d0-timpk
          go to 129
        end if
c
      end if
      ppow=energ/30.0d0
      write (2, 311) ppow/cur,cur,ppow,vv
      if(ppow.gt.pwrpmax .and. vv.gt.vcut) then
        if(curpmax.lt.cur) then
          curmin=curpmax
          pwrmin=pwrpmax
          vfmin=vvpmax
        else
          curmax=curpmax
          pwrmax=pwrpmax
          vfmax=vvpmax
        endif
        curpmax=cur
        pwrpmax=ppow
        vvpmax=vv
endif
      if(vv.lt.vcut) then
        if(curmax.eq.0.d0 .or. cur.lt.curmax) then
          curmax=cur
          pwrmax=ppow
          vfmax=vv
        endif
      else
        if(cur.gt.curmin .and. cur.lt.curpmax) then
          curmin=cur
          pwrmin=ppow
          vfmin=vv
        endif
        if(cur.gt.curpmax) then
          curmax=cur
          pwrmax=ppow
          vfmax=vv
        endif
      endif
      if(curmax.eq.0.0d0) go to 128
      if(curmin. lt. 0.999d0*curmax) go to 128
      write (2, 311) pwrpmax/curpmax,curpmax,pwrpmax,vvpmax
write (2, *) ' finished with vcut= ',vcut
      if(vcut.eq.0.0d0) return
      vcut=0.0d0
      go to 127
c
      end
c*********************************************************************
      subroutine cellpot(kk,v,li,lpow,lflag)
      implicit real*8(a-h,o-z)
      parameter(maxt=900)
      common /n/ nx,nt,n1,n2,nj,n3,tmmax
      common /calc/ ai(maxt),ai2(maxt),u(223,maxt),ts(maxt),
     &    h,h1,h2,h3,hcn,hcp,rr,rrmax
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     &    ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1,ep(221),
     &    epp(221),epf(221)
      common/power/ ed,Vold,ranode,rcathde
      common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
      common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,sig(221),area(221),
     &    sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
      common/temp/ thk,htc,dudt,Cp,dens,tam,g0,ncell,lht,qq
 309 format(f8.5,', ',f8.5,', ',f8.3,', ',g11.6,',',' ',f9.5,', ',f7.3,
 &    ',',g10.5)
 307 format(f8.5,', ',f8.5,', ',f9.5,', ',g11.6,',','  ',f7.3,',',f8.3)
c
c    Material balance criteria:
      sum=0.0d0
      w=0.0d0
c
c
c     if (n1 .gt. 2) then
c       do 85 j=2,n1
c  85    sum=sum+xt(1,j,kk)*(ep(j)+epp(j))*h1
c     endif
c     sum=sum+(xt(1,1,kk)+xt(1,n1+1,kk))*(ep(j)+epp(j))*h1/2.0d0
c     do 86 j=n1+2,n2+n1-1
```

```
c  86 sum=sum+xt(1,j,kk)*(ep(j)+epp(j))*h2
c     sum=sum+(xt(1,n1+1,kk)+xt(1,n2+n1,kk))*(ep(j)+epp(j))*h2/2.0d0
c     do 87 j=n2+n1+1,nj-1
c  87 sum=sum+xt(1,j,kk)*(ep(j)+epp(j))*h3
c     sum=sum+(xt(1,n1+n2,kk)+xt(1,nj,kk))*h3*(ep(nj)+epp(nj))/2.0d0
c    calculate total salt in cell from initial profile:
c     w=xt(1,n1+2,1)*((n2-1)*(ep2+epp2)*h2+n1*(ep1+epp1)*h1
c    &     +n3*(ep3+epp3)*h3)
if(lflag.eq.0) w=w-(xt(1,n1+2,1)-xt(1,1,1))*(n1*(ep1+epp1)*h1
c    &     +n3*(ep3+epp3)*h3)
c     if(lflag.eq.0) w=w-(xt(1,n1+2,1)-xt(1,1,1))*(ep2+epp2)*h2
c    material balance parameter should be ca=1.00
c
c
c    NEW MATERIAL BALANCE
c
c
      if (n1 .gt. 2) then
        sum=sum+xt(1,1,kk)*(ep(1)+epp(1))*h1/2.0d0
c
        do 85 j=2,n1
          sum=sum+xt(1,j,kk)*(ep(j)+epp(j))*h1
  85    continue
c
        sum=sum+xt(1,n1+1,kk)*(ep(n1+1)+epp(n1+1))*h1/2.0d0
      endif
        sum=sum+xt(1,n1+1,kk)*(ep(n1+2)+epp(n1+2))*h2/2.0d0
c
      do 86 j=n1+2,n1+n2-1
      sum=sum+xt(1,j,kk)*(ep(j)+epp(j))*h2
86 continue
c
      sum=sum+xt(1,n1+n2,kk)*(ep(n1+n2)+epp(n1+n2))*h2/2.0d0
      sum=sum+xt(1,n1+n2,kk)*(ep(n1+n2+1)+epp(n1+n2+1))*h3/2.0d0
c
      do 87 j=n1+n2+1,nj-1
        sum=sum+xt(1,j,kk)*(ep(j)+epp(j))*h3
  87 continue
c
      sum=sum+xt(1,nj,kk)*(ep(nj)+epp(nj))*h3/2.0d0
c
c    Calculate total amount of salt in cell
c
      if (n1 .gt.2) then
        w=w+xt(1,1,1)*(ep(1)+epp(1))*h1/2.0d0
c
```

```
       do 88 j=2,n1
         w=w+xt(1,j,1)*(ep(j)+epp(j))*h1
  88   continue
c
       w=w+xt(1,n1+1,1)*(ep(n1+1)+epp(n1+1))*h1/2.0d0
       endif
w=w+xt(1,n1+1,1)*(ep(n1+2)+epp(n1+2))*h2/2.0d0
c
     do 89 j=n1+2,n1+n2-1
       w=w+xt(1,j,1)*(ep(j)+epp(j))*h2
  89 continue
c
     w=w+xt(1,n1+n2,1)*(ep(n1+n2)+epp(n1+n2))*h2/2.0d0
     w=w+xt(1,n1+n2,1)*(ep(n1+n2+1)+epp(n1+n2+1))*h3/2.0d0
c
     do 90 j=n1+n2+1,nj-1
       w=w+xt(1,j,1)*(ep(j)+epp(j))*h3
  90 continue
c
     w=w+xt(1,nj,1)*(ep(nj)+epp(nj))*h3/2.0d0
c
c
c
     ca=sum/w
c
c
     if (kk.eq.1) then
       ut=xt(3,nj,1)/ct3
ut2=xt(3,1,1)/ct1
     end if
c
c     Calculate cell potential from dif of solid phase potentials:
     v=xt(6,nj,kk)-xt(6,1,kk)
c
c     CALCULATE CAPACITY OF ANODE AND CATHODE
c
     cathcap=0.0d0
     cathcap=cathcap+(1.0d0-ep(n1+n2+1)-epp(n1+n2+1)-
    &     epf(n1+n2+1))*h3*ct3/2.0d0
     do 105 jc=n1+n2+1,nj-1
     cathcap=cathcap+(1.0d0-ep(jc)-epp(jc)-epf(jc))*h3*ct3
  105 continue
     cathcap=cathcap+(1.0d0-ep(nj)-epp(nj)-epf(nj))*h3*ct3/2.0d0

     ancap=0.0d0
     ancap=ancap+(1.0d0-ep(1)-epp(1)-epf(1))*
```

```
   &    h1*ct1/2.0d0
      do 106 jc=2,n1
      ancap=ancap+(1.0d0-ep(jc)-epp(jc)-epf(jc))*h1*ct1
  106 continue
ancap=ancap+(1.0d0-ep(n1+1)-epp(n1+1)-epf(n1+1))*h1*
   &    ct1/2.0d0

c     Calculate utilization of two electrodes based on coulombs passed:
      if(li.eq.1) then
c     Calculate energy density by running sum of currentxvoltage:
        ed=ed+(Vold+v)*(ts(kk)-ts(kk-1))*cur/2.0d0
        Vold=v
        ut=cur*(ts(kk)-ts(kk-1))/fc/cathcap+ut
        if (n1 .gt. 0)        !***Need to fix how utilization is calculated fo
r a foil anode
   &       ut2=ut2-cur*rr/fc/ancap
        th=ts(kk)/6.0d01
c
        if(lht.ne.2) call temperature(kk,v,Uoc,Soc)
        tprint=t-273.15
        if(lpow.ne.0) then
c     ! isothermal peak power output:
          write (2,309) v,ca,cur,v*cur
        else
          if (lht.eq.0) then  ! T varies, uses htc:
            write (2,309) ut,v,tprint,th,Uoc,cur,qq
else if (lht.eq.1) then ! calculated htc:
            write (2,309) ut,v,htc,th,Uoc,cur,qq
          else if (lht.eq.2) then ! isothermal output:
c     write (2, 307) ut,v,ca,th,cur,kk,ed
            write (2,307) ut,v,ca,th,cur,ed/tw/3.6d3
          endif
        endif
      endif
      jref=(n1+1+n1+n2)/2
  310 format (1p3e20.6)
      write (3,310) th,xt(6,1,kk)-xt(2,jref,kk)
     %     ,xt(6,nj,kk)-xt(2,jref,kk)
c
      return
      end
c*********************************************************************
      subroutine sol(nmax,jj)
      implicit real*8(a-h,o-z)
c     This subroutine calculates the solid phase concentration profiles.
      parameter(maxt=900)
      common /calc/ ai(maxt),ai2(maxt),u(223,maxt),ts(maxt),
     &     h,h1,h2,h3,hcn,hcp,rr,rrmax
  common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     &     ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1,ep(221),
     &     epp(221),epf(221)
      common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,sig(221),area(221),
     &     sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
      dimension cs(50)
c
c     set initial value of solid concentration
      do 88 i=1, 50
c     cs(i)=0.0d0
   88 cs(i)=xt(3,jj,1)
c
c     complete calculations for 50 points along radius of particle
      nmax=nmax-1               ! added
      do 10 i=1,50
        y2=0.02d0*dble(i)
c
        sum1=0.0d0
        do 20 kk=1,nmax
          k=nmax+1-kk
c
t1=(ts(nmax+1)-ts(k))*dfs1/Rad1/Rad1
          sum2=sum1
c
c     calculate c bar (r,t1)
          sum1=0.0d0
          r1=1.0d0
c
          do 89 j=1,15
            r1=-r1
            y1=j*j*pi*pi*t1
            y3=j*pi*y2
            if (y1 .gt. 1.50d02) then
              da=0.0d0
            else
              da=expf(-y1)
            end if
   89       sum1=sum1-2.0d0*r1*da*dsin(y3)/j/pi/y2
          sum1=1.0d0-sum1
c
c     perform superposition
c
          cs(i)=cs(i)+(xt(3,jj,k+1)+xt(3,jj,k)-2.0d0*xt(3,jj,1)
&        )*(sum1-sum2)/2.0d0
 20   continue
c
 10 continue
   nmax=nmax+1              ! added
c
   write (2,*) ' '
   write (2,*) 'time is ',ts(nmax)
   write (2,*) ' '
   do 90 i=1, 50, 1
 90 write (2,*) .02*i,' ',cs(i)
c
   return
   end
c*****************************************************************************
   subroutine mass(re,rs3,rs1,rf,rpl,rc,rcn,rcp)
   implicit real*8(a-h,o-z)
   parameter(maxt=900)
   common /n/ nx,nt,n1,n2,nj,n3,tmmax
   common /calc/ ai(maxt),ai2(maxt),u(223,maxt),ts(maxt),
  &    h,h1,h2,h3,hcn,hcp,rr,rrmax
   common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
&    ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1,ep(221),
  &    epp(221),epf(221)
   common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
   common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,sig(221),area(221),
  &    sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
c
c   DOYLE MASS CALCULATION
c
c   mass of positive electrode
c    c1=h3*n3*(re*ep3+rpl*epp3+rs3*(1.0d0-ep3-epf3-epp3)+rf*epf3)
c
c   mass of separator
c    s=(re*ep2+rpl*epp2+rc*(1-ep2-epp2))*h2*(n2-1)
c
c   mass of negative electrode
c    n1hold = n1
c    if (n1 .eq. 0) n1hold = 1
c    a1=h1*n1hold*(re*ep1+rpl*epp1+rs1*(1.0d0-ep1-epf1-epp1)+rf*epf1)
c
c   mass of current collectors
   cc1=rcn*hcn+rcp*hcp
c
c   ******************************
c   New Mass Calculations
```

```
c    ********************************
c

      c1=0.0d0
      s=0.0d0
      a1=0.0d0
c
c
      if (n1 .eq. 0) then
        a1=h1*(re*ep1+rpl*epp1+rs1*(1.0d0-ep1-epf1-epp1)+rf*epf1)
      else
        a1=a1+h1*(re*ep(1)+rpl*epp(1)+rs1*
     &      (1.0d0-ep(1)-epf(1)-epp(1))+rf*epf(1))/2.0d0
        do 50 j=2,n1
          a1=a1+h1*(re*ep(j)+rpl*epp(j)+rs1*
     &        (1.0d0-ep(j)-epf(j)-epp(j))+rf*epf(j))
 50     continue
        a1=a1+h1*(re*ep(n1+1)+rpl*epp(n1+1)+rs1*(1.0d0-ep(n1+1)-
     &      epf(n1+1)-epp(n1+1))+rf*epf(n1+1))/2.0d0
      endif
s=s+h2*(re*ep(n1+2)+rpl*epp(n1+2)+rc*(1.0d0-ep(n1+2)-
     &    epp(n1+2)))/2.0d0
      do 51 j=n1+2,n1+n2-1
        s=s+h2*(re*ep(j)+rpl*epp(j)+rc*(1.0d0-ep(j)
     &      -epp(j)))
 51   continue
      s=s+h2*(re*ep(n1+n2)+rpl*epp(n1+n2)+rc*(1.0d0-
     &    ep(n1+n2)-epp(n1+n2)))/2.0d0
      c1=c1+h3*(re*ep(n1+n2+1)+rpl*epp(n1+n2+1)+rs3*(1.0d0-
     &    ep(n1+n2+1)-epf(n1+n2+1)-epp(n1+n2+1))+rf*
     &    epf(n1+n2+1))/2.0d0
      do 52 j=n1+n2+1,nj-1
        c1=c1+h3*(re*ep(j)+rpl*epp(j)+rs3*(1.0d0-ep(j)
     &      -epf(j)-epp(j))+rf*epf(j))
 52   continue
      c1=c1+h3*(re*ep(nj)+rpl*epp(nj)+rs3*(1.0d0-ep(nj)-
     &    epf(nj)-epp(nj))+rf*epf(nj))/2.0d0
c
c    *******
c    New Mass Calculations done
c    *******
c
 tw=c1+s+a1+cc1
c
      return
      end
```

```
c**********************************************************************
      subroutine temperature(kk,v,Uoc, Soc)
      implicit real*8(a-h,o-z)
      parameter(maxt=900)
      common /n/ nx,nt,n1,n2,nj,n3,tmmax
      common /calc/ ai(maxt),ai2(maxt),u(223,maxt),ts(maxt),
     &    h,h1,h2,h3,hcn,hcp,rr,rrmax
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     &    ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1,ep(221),
     &    epp(221),epf(221)
      common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
      common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,sig(221),area(221),
     &    sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
      common/temp/ thk,htc,dudt,Cp,dens,tam,g0,ncell,lht,qq
c
c
c     Revised by Karen Thomas August 5, 1999 to calculated the
c     enthalpy potential as an average weighted by the local
c     reaction rate.
c     The entropy and open circuit potential for each electrode
c     should be given in ekin with respect to a Li reference electrode
c     at the same local electrolyte concentration.
c     Caution in using Uoc: it does not have units of volts
c     until the last line of this subroutine.
c     If heat from side reactions is to be included, add the term
c     reaction rate*enthalpy of reaction inside the summation at
c     each electrode.
c     Heat generation is negative if exothermic.
c     The time stepping used to calculate the new temperature has been
c     modified, so that the new temperature changes due to heat
c     generated or exchanged at the old temperature.
c     If cur = 0, the Uoc = sum(U*local reaction rate)=0 unless the
c     cell is relaxing from a previous charge or discharge.  If Uoc = 0,
c     then v is the open circuit potential.
c
c     Negative Electrode
c
       call ekin(1,kk,0,0)
       Ua=-g0*fc*area(1)*h1*xx(5,1)
       Sa =-dudt*fc*area(1)*h1*xx(5,1)
 if (n1 .gt. 1) then
         Ua = 0.0
         Sa = 0.0
         sum =0.0
         do 868 j = 1, n1+1, 1
           call ekin(j,kk,0,0)
```

```
          trap = 1.0         !factor for trapezoidal integration
          if ((j .eq. 1) .or. (j .eq. n1+1)) trap = 0.5
          h = h1
          if (j .eq. n1+1) h = h2
          Ua=Ua-trap*fc*area(j)*h*xx(5,j)*g0 !negative sign needed for reactio
n rate
          Sa=Sa-trap*fc*area(j)*h*xx(5,j)*dudt
 868      continue
      endif
c
c     Positive Electrode
c
      Uc = 0.0
      Sc = 0.0
      do 878 j = n1+n2,nj,1
        call ekin(j,kk,0,0)
trap = 1.0
        if ((j .eq. n1+n2) .or. (j .eq. nj)) trap = 0.5
        Uc=Uc-trap*fc*area(j)*h3*xx(5,j)*g0
        Sc=Sc-trap*fc*area(j)*h3*xx(5,j)*dudt
 878 continue
c
      Uoc = Uc+Ua              !add because signs different on reaction rate
      Soc = Sc+Sa
c
c     Per cell heat generation
c
      qq=cur*v - Uoc +t*Soc     ! heat is negative if exothermic
c
c     The heat transfer coefficient is for heat transferred out of
c     one side of the cell; it is defined based on cell area.
c     htcc is a per-cell heat transfer coefficient.
c
      if (lht.eq.0) then        !cell temperature changes
        htcc=htc/Ncell
        t=t+(rr/(dens*Cp*thk))*(htcc*(tam-t)-cur*v+Uoc-t*Soc) !note change in t
ime derivative
      else
c
c     Calculate htc instead of temperature: the heat transfer coefficient
c     required to keep the temperature constant is
c     calculated as a function of time.  The heat transfer coef.
c     is calculated for heat transferred out of one side of the
c     cell stack.  Htcc is defined as a per-cell heat transfer
c     coefficient.
c
```

```
        if (t.ne.tam) then
          htc=Ncell*(Uoc-v*cur-t*Soc)/(t-tam)
        else
          htc=0.0
        endif
        htcc=htc/Ncell
      endif
c
      if (dabs(cur) .gt. 0.0) then
        Uoc = Uoc/cur
        Soc = Soc/cur
      endif
      return
      end
c*******************************************************************
      double precision function expf(x)
      implicit real*8 (a-h,o-z)
      expf=0.d0
      if(x.gt.-700.d0) expf=dexp(x)
      return
      end
c*******************************************************************
      subroutine ekin(j,kk,lag,utz)
      implicit real*8(a-h,o-z)
c     This subroutine evaluates the Butler-Volmer equations.
      parameter(maxt=900)
      common /n/ nx,nt,n1,n2,nj,n3,tmmax
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     &     ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1,ep(221),
     &     epp(221),epf(221)
      common/power/ ed,Vold,ranode,rcathde
      common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
      common/cprop/ sig3,area3,rka3,ct3,dfs3,Rad3,sig(221),area(221),
     &     sig1,area1,rka1,ct1,dfs1,Rad1,tw,cap1,cap3
      common/temp/ thk,htc,dudt,Cp,dens,tam,g0,ncell,lht,qq
      common/mat/ b,d
common/bnd/ a,c,g,x,y
      dimension b(10,10),d(10,21)
      dimension a(10,10),c(10,221),g(10),x(10,10),y(10,10)
c
c     Calculate average open-circuit potential in either
c     electrode if lag=1, otherwise lag=0
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%
c
```

```
c    OPEN-CIRCUIT POTENTIAL FUNCTIONS:
c
c    g0 is the open-circuit potential in terms of the solid
c    concentration, xx(3,j), with respect to a lithium metal
c    electrode
c    g1 is the derivative of the open-circuit potential wrt
c    the solid concentration
c
c
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&
c
c    FOR THE NEGATIVE ELECTRODE
c
     if (j .le. n1+1) then
c
       if (lag.eq.1) xx(3,j)=utz*ct1
c
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c
       go to (51,52,53,54,55,56),nneg
  51   go to 111             ! Li foil
  52   go to 112             ! Carbon (petroleum coke)
  53   go to 113             ! MCMB 2510 Carbon (Bellcore)
  54   go to 114             ! TiS2
  55   go to 115             ! Tungsten oxide (LixWO3 with 0<x<0.67)
  56   go to 116             ! Lonza KS6 graphite (Bellcore)
c
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    Li Foil (works only if n1 = 0)
c
 111   g0 = 0.0
       g1 = 0.0
c
       go to 97
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    Carbon (petroleum coke):
c
 112   c1=-0.132056d0
       c2=1.40854d0
       c3=-3.52312d0
       g0=c1+c2*expf(c3*xx(3,j)/ct1)
       g1=c2*c3*expf(c3*xx(3,j)/ct1)/ct1
       go to 97
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    MCMB 2510 carbon (Bellcore)
```

```
c
 113   c1=-0.160d0
       c2=1.32d0
       c3=-3.0d0
       g0=c1+c2*expf(c3*xx(3,j)/ct1)
       g0=g0+10.d0*expf(-2000.d0*xx(3,j)/ct1)
       g1=c2*c3*expf(c3*xx(3,j)/ct1)/ct1
       g1=g1-10.d0*2000.d0/ct1*expf(-2000.d0*xx(3,j)/ct1)
       go to 97
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    TiS2
c
 114   delt=-5.58d-04
  zeta=8.1d0
 c    ct1=2.9d04
        g0=2.17+(dlog((ct1-xx(3,j))/xx(3,j))+delt*xx(3,j)+zeta)/frt
        g1=(delt-ct1/xx(3,j)/(ct1-xx(3,j)))/frt
        go to 97
 c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
 c    Tungsten oxide (LixWO3 with 0<x<0.67):
 c    literature data from Whittingham
 c
  115   c1=2.8767d0
        c2=-0.9046d0
        c3=0.76679d0
        c4=-0.15975d0
        c5=0.671d0
        g0=c1+c2*xx(3,j)/ct1+c3*xx(3,j)*xx(3,j)/ct1/ct1+
     &       c4*expf(100.0d0*(xx(3,j)/ct1-0.671))
        g1=c2/ct1+2.0d0*c3*xx(3,j)/ct1/ct1+
     &       c4*100.0d0*expf(100.0d0*(xx(3,j)/ct1-0.671))/ct1
        go to 97
 c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
 c    Bellcore graphite (Lonza KS6)
 c
 116   c1=0.7222d0
        c2=0.13868d0
        c3=0.028952d0
        c4=0.017189d0
        c5=0.0019144d0
        c6=0.28082d0
        c7=0.79844d0
        c8=0.44649d0
        xtem=xx(3,j)/ct1
        g0=c1+c2*xtem+c3*(xtem**0.5d0)-c4*(xtem**-1.0d0)
     &       +c5*(xtem**-1.5d0)+c6*expf(15.0d0*(0.06d0-xtem))-c7
```

```
   &      *expf(c8*(xtem-0.92d0))
      g1=c2/ct1+c3*0.5d0*(ct1**-0.5d0)*(xx(3,j)**-0.5d0)
   &+c4*ct1*(xx(3,j)**-2.0d0)-c5*1.5d0*(ct1**1.5d0)*(xx(3,j)**-2.5d0)
   &-c6*15.0d0/ct1*expf(15.0d0*(0.06d0-xtem))
   &-c7*c8/ct1*expf(c8*(xtem-0.92d0))
      go to 97
c
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&
c
c   KINETIC EXPRESSIONS FOR NEGATIVE ELECTRODE
c
c   h0 is the exchange current density (A/m2)
c   h1 is the derivative of io wrt solid concentration, xx(3,j)
c   h2 is the derivative of io wrt electrolyte concen., xx(1,j)
c
c   &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c
c   NONAQUEOUS LIQUIDS
c
  97   if (lag.eq.1) go to 99
       alpha=0.5d0
       alphc=0.5d0
       if (n1 .eq. 0 .and. j .lt. n1+2) then
c    alpha = 0.67 for Li.  But for now, sticking with .5.
c    h0=rka1*(xx(1,j)**0.67)
         h0=rka1*dsqrt(xx(1,j))
         h1 = 0.0d0
         h2 = rka1/dsqrt(xx(1,j))/2.0d0
c    h2 = rka1*(xx(1,j)**(-.33))*0.67
       else
         h0=rka1*dsqrt(xx(1,j))*dsqrt(ct1-xx(3,j))*dsqrt(xx(3,j))
         h1=-rka1*dsqrt(xx(1,j))*dsqrt(ct1-xx(3,j))*dsqrt(xx(3,j))*(1.0d0
    &     /(ct1-xx(3,j))-1.0d0/xx(3,j))/2.0d0
         h2=rka1*dsqrt(ct1-xx(3,j))*dsqrt(xx(3,j))/dsqrt(xx(1,j))/2.0d0
endif
c
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c
c    POLYMER
c
c    cmax=3.92d03
c    alpha=0.5d0
c    alphc=0.5d0
c    h0=rka1*dsqrt(xx(1,j))*dsqrt(cmax-xx(1,j))*dsqrt(ct1-xx(3,j))
c    1*dsqrt(xx(3,j))
```

```
c    h1=-rka1*dsqrt(xx(1,j))*dsqrt(cmax-xx(1,j))*dsqrt(ct1-xx(3,j))
c     1*dsqrt(xx(3,j))*(1.0d0/(ct1-xx(3,j))-1.0d0/xx(3,j))/2.0d0
c    h2=-rka1*dsqrt(xx(3,j))*dsqrt(ct1-xx(3,j))*dsqrt(cmax-xx(1,j))
c     1*dsqrt(xx(1,j))*(1.0d0/(cmax-xx(1,j))-1.0d0/xx(1,j))/2.0d0
c
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c
      end if
c
c
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&
c
c    FOR THE POSITIVE ELECTRODE
c
      if (j .ge. n1+n2) then
c
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c
        if (lag.eq.1) xx(3,j)=utz*ct3
        go to (1,2,3,4,5,6,7,8,9,10,11),npos
    1   go to 201            ! TiS2
    2   go to 202            ! Spinel Mn2O4 (lower plateau)
    3   go to 203            ! NaCoO2:  Sodium cobalt oxide
    4   go to 204            ! Spinel Mn2O4 (upper plateau)
    5   go to 205            ! Tungsten oxide (LixWO3 with 0<x<0.67)
    6   go to 206            ! CoO2 (Cobalt dioxide)
    7   go to 207            ! V2O5 (Vanadium oxide)
    8   go to 208            ! NiO2 (Nickel dioxide)
    9   go to 209            ! Spinel Mn2O4 (Bellcore)
   10    go to 210            ! V6O13 (Vanadium oxide)
   11    go to 211            ! LiAl0.2Mn1.8O4F0.2 spinel (Bellcore)
c
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    TiS2
c
  201   delt=-5.58d-04
        zeta=8.1d0
        g0=2.17+(dlog((ct3-xx(3,j))/xx(3,j))+delt*xx(3,j)+zeta)/frt
        g1=(delt-ct3/xx(3,j)/(ct3-xx(3,j)))/frt
        go to 98
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    Spinel Mn2O4 (lower plateau)
c
  202   c1=2.06307d0
        c2=-0.869705d0
      c3=8.65375d0
      c4=0.981258d0
      a1=c3*(xx(3,j)/ct3-c4)
      g0=c1+c2*(dtanh(a1))
      g1=c2*c3/ct3/(dcosh(a1))/(dcosh(a1))
      go to 98
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    NaCoO2:  Sodium Cobalt Oxide (P2 phase, 0.3<y<0.92)
c
 203  c1=4.4108d0
      c2=-2.086d0
 c3=0.10465d0
      c4=133.42d0
      c5=89.825d0
      c6=0.16284d0
      c7=145.01d0
      c8=71.92d0
      c9=0.01d0
      c10=200.0d0
      c11=0.3d0
      c12=0.885d0
      a1=xx(3,j)/ct3
      g0=c1+c2*a1+c3*dtanh(-c4*a1+c5)+c6*dtanh(-c7*a1+c8)+c9*
     &  expf(-c10*(a1-c11))-c9*expf(c10*(a1-c12))
      g1=c2/ct3-c3*c4/dcosh(-c4*a1+c5)/dcosh(-c4*a1+c5)/ct3-c6*c7
     &/dcosh(-c7*a1+c8)/dcosh(-c7*a1+c8)/ct3-c9*c10*expf(-c10*(a1-c11))/ct3
     &-c9*c10*expf(c10*(a1-c12))/ct3
      go to 98
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    Spinel Mn2O4 (upper plateau 0.2<y<1.0)
c    Literature version
c
 204  a1=4.06279d0
 a2=0.0677504d0
        a3=21.8502d0
        a4=12.8268d0
        a5=0.105734d0
        a6=1.00167d0
        a7=-0.379571d0
        a8=1.575994d0
        a9=0.045d0
        a10=71.69d0
        a11=0.01d0
        a12=200.0d0
        a13=0.19d0
        if(xx(3,j).gt.a6*ct3) write (2, *) 'xx(3,j)>a6 at j=',j
      g0=a1+a2*dtanh(-a3*xx(3,j)/ct3+a4)-a5*((a6-xx(3,j)/ct3)**a7-
    &      a8)-a9*expf(-a10*((xx(3,j)/ct3)**8.0d0))+a11
    &      *expf(-a12*(xx(3,j)/ct3-a13))
      g1=(1.0d0/ct3)*(-a2*a3/dcosh(-a3*xx(3,j)/ct3+a4)/dcosh(-a3
    &  *xx(3,j)/ct3+a4)+a5*a7*(a6-xx(3,j)/ct3)**(-1.0d0+a7)+
    &  a9*a10*8.0d0*((xx(3,j)/ct3)**7.0d0)*expf(-a10*
    &  (xx(3,j)/ct3)**8.0d0))-a11*a12/ct3*expf(-a12*(xx(3,j)/ct3-a13))
c
      if (g0.gt.4.5) then
   g0=4.5d0
        g1=0.0d0
c    write (2,*) 'U theta overflow - positive'
      else if (g0.lt.3.0) then
        g0=3.0d0
        g1=0.0d0
c    write (2,*)'U theta underflow - positive'
      end if
      go to 98
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    Tungsten oxide (LixWO3 with 0<x<0.67)
c    literature data from Whittingham
c
 205    c1=2.8767d0
      c2=-0.9046d0
      c3=0.76679d0
      c4=-0.15975d0
      c5=0.671d0
      g0=c1+c2*xx(3,j)/ct3+c3*xx(3,j)*xx(3,j)/ct3/ct3+
    &      c4*expf(100.0d0*(xx(3,j)/ct3-0.671))
      g1=c2/ct3+2.0d0*c3*xx(3,j)/ct3/ct3+
    &      c4*100.0d0*expf(100.0d0*(xx(3,j)/ct3-0.671))/ct3
   go to 98
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    CoO2 (Cobalt dioxide)
c
 206    r1=4.825510
      r2=0.950237
      r3=0.913511
      r4=0.600492
      g0=r1-r2*expf(-((xx(3,j)/ct3-r3)/r4)**2.0d0)
      g1=2.0d0*r2*(xx(3,j)/ct3-r3)*expf(-((xx(3,j)/ct3
    &      -r3)/r4)**2.0d0)/r4/r4/ct3
      go to 98
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    V2O5 (Vanadium oxide 0<y<0.95)
c
```

```
 207   r1=3.3059d0
       r2=0.092769d0
       r3=14.362d0
       r4=6.6874d0
       r5=0.034252d0
       r6=100.0d0
       r7=0.96d0
 r8=0.00724d0
       r9=80.0d0
       r10=0.01d0
       a2=xx(3,j)/ct3
       g0=r1+r2*dtanh(-r3*a2+r4)-r5*expf(r6*(a2-r7))+r8*expf(r9*
     &       (r10-a2))
       g1=-r2*r3/dcosh(-r3*a2+r4)/dcosh(-r3*a2-r4)/ct3-r5*r6*expf(r6*
     &       (a2-r7))/ct3-r8*r9*expf(r9*(r10-a2))/ct3
       go to 98
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    NiO2 (Nickel dioxide 0.45<y<1.0)
c
 208  r1=6.515d0
      r2=2.3192d0
      r3=5.3342d0
      r4=0.41082d0
      r5=200.0d0
      r6=0.44d0
      r7=0.24247d0
      r8=60.0d0
      r9=0.99d0
      a3=xx(3,j)/ct3
g0=r1+r2*a3-r3*a3**0.5d0+r4*expf(r5*(r6-a3))-r7*expf(r8*(a3-r9))
      g1=r2/ct3-0.5d0*r3*(xx(3,j)**-0.5d0)/ct3**0.5d0
     &        -r4*r5*expf(r5*(r6-a3))/ct3-r7*r8*expf(r8*(a3-r9))/ct3
       go to 98
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    Spinel Mn2O4 (Bellcore 0.17<y<1.0)
c
 209   a1=4.19829d0
       a2=0.0565661d0
       a3=14.5546d0
       a4=8.60942d0
       a5=0.0275479d0
       a6=0.998432d0         ! would prefer this to be >=1
       a7=-0.492465d0
       a8=1.901110d0
       a9=0.157123d0
       a10=0.04738d0
      a11=0.810239d0
      a12=40.0d0
      a13=0.133875d0
cccc  if(xx(3,j).gt.a6*ct3) write (2,*) '#109 in ekin, j=',j
c    g0=a1+a2*dtanh(-a3*xx(3,j)/ct3+a4)-a5*((a6-xx(3,j)/ct3)**a7-
c    1a8)-a9*expf(-a10*((xx(3,j)/ct3)**8.0d0))+a11
c    1*expf(-a12*(xx(3,j)/ct3-a13))
      g0=a1+a2*dtanh(-a3*xx(3,j)/ct3+a4)
     &      -a9*expf(-a10*((xx(3,j)/ct3)**8.0d0))+a11
     &      *expf(-a12*(xx(3,j)/ct3-a13))+a5*a8
      if(xx(3,j).lt.a6*ct3) g0=g0-a5*((a6-xx(3,j)/ct3)**a7)

c    g1=(1.0d0/ct3)*(-a2*a3/dcosh(-a3*xx(3,j)/ct3+a4)/dcosh(-a3
c    1*xx(3,j)/ct3+a4)+a5*a7*(a6-xx(3,j)/ct3)**(-1.0d0+a7)+
c    1a9*a10*8.0d0*((xx(3,j)/ct3)**7.0d0)*expf(-a10*
c    1(xx(3,j)/ct3)**8.0d0))-a11*a12/ct3*expf(-a12*(xx(3,j)/ct3-a13))
      g1=(1.0d0/ct3)*(-a2*a3/dcosh(-a3*xx(3,j)/ct3+a4)/dcosh(-a3
     &  *xx(3,j)/ct3+a4)
     &  +a9*a10*8.0d0*((xx(3,j)/ct3)**7.0d0)*expf(-a10*
     &  (xx(3,j)/ct3)**8.0d0))-a11*a12/ct3*expf(-a12*(xx(3,j)/ct3-a13))
     if(xx(3,j).lt.a6*ct3)g1=g1+a5*a7*(a6-xx(3,j)/ct3)**(-1.0d0+a7)/ct3

cccc  if(xx(3,j).gt.a6*ct3) write (2, *) 'did it'
c
      if (g0.gt.6.0) then
        g0=6.0d0
        g1=0.0d0
c    write (2,*) 'U theta overflow - positive'
      else if (g0.lt.3.0) then
        g0=3.0d0
        g1=0.0d0
c    write (2,*) 'U theta underflow - positive'
      end if
      go to 98
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
c    Nonstoichiometric Vanadium oxide (V6O13)
c    based on data from West, Zachau-Christiansen, and Jacobsen,
c    Electrochim. Acta vol 28, p. 1829, 1983.
c    valid for 0.1 < x < 8.25 in LixV6O13.  Enter csx according to
c    LiyVO2.167, where 0.05 <y <1, and cot3 is based on 8 Li inserted.
c    Fit for electrical conductivity based on data from same
c    paper, corrected for porosity.  Electrical conductivity
c    of V6O13- carbon filler composite based on model of Meredith and
c    Tobias in Advances in Electrochemistry and Electrochemical Engineering
c    vol. 2, 1962.
```

```
 210   continue
c    convert solid concentration in mol/m3 to x in mol/mol V6O13
       sto = xx(3,j)*8.0/ct3
a1 = -10.0/6.0
       a2 = 8.0/6.0/ct3
       g0 = 1.9 + 0.13*dtanh(a1*sto + 1.7)
    &       + 0.2*dtanh(sto*a1 + 6.2)
    &       + 0.56*dtanh(2.0*a1*sto + 29.5)
       g1=-(1.3*a2)/(dcosh(a1*sto+1.7)*dcosh(a1*sto+1.7))
    &       -(2.0*a2)/(dcosh(a1*sto+6.2)*dcosh(a1*sto+6.2))
    &       -(11.2*a2)/(dcosh(2.0*a1*sto+29.5)
    &       *dcosh(2.0*a1*sto+29.5))
c    conductivity varies with state of charge for this material
       vanox=120.0*expf(-1.5*sto)-9.0*dtanh(1.5*sto-6.5)+9.0
       sigcarb = 100.0/0.0038 ! S/m
       boo = sigcarb/vanox
       sig(j)=vanox*(2.0d0*(boo+2.0d0)+2.0d0*(boo - 1.0)*epf3)
    &       *((2.0-epf3)*(boo + 2.0)+2.0*(boo-1.0)*epf3)/
    &       (2.0*(boo+2.0)-(boo-1.0)*epf3)/
    &       ((2.0-epf3)*(boo+2.0)-(boo-1.0)*epf3)
       sig(j) = sig(j)*(1.0d0 - ep3 - epp3)**1.5
       go to 98
c
c
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    LiyAl0.2Mn1.8O4F0.2 spinel from Bellcore.  Open circuit potential
c    as function of y measured by Karen Thomas, 1999.
c    for 0.2 < y < 0.95
 211   continue
c
       a1= 3.91007
       a2= 0.04697
       a3= 9.15495
       a4= 5.35279
       a5= 0.06752
       a6= 1.0179
       a7=-0.471987
       a8= 14.7524
       a9= 0.69465
       a10= 0.21481
       a11= 0.0008189
       a12= 22.759
       a13= 1.0 -0.631724
       a14= 26.558
c
```

```
      g0=a1+a2*dtanh(-a3*xx(3,j)/ct3+a4)
     &      -a9*expf(-a10*((xx(3,j)/ct3)**a14))+a11
     &      *expf(-a12*(xx(3,j)/ct3-a13))+a5*a8
     if(xx(3,j).lt.a6*ct3)
     &      g0=g0-a5*((a6-xx(3,j)/ct3)**a7)
c
      g1=(1.0d0/ct3)*(-a2*a3)/dcosh(-a3*xx(3,j)/ct3+a4)/
     &      dcosh(-a3*xx(3,j)/ct3+a4)
     &      +a9/ct3*a10*a14*((xx(3,j)/ct3)**(a14-1.0d0))
     &      *expf(-a10*((xx(3,j)/ct3)**a14))
     &      -a11*a12/ct3*expf(-a12*(xx(3,j)/ct3-a13))
      if(xx(3,j).lt.a6*ct3)
     &      g1=g1+a5*a7*(a6-xx(3,j)/ct3)**(-1.0d0+a7)/ct3
c
      if (g0.gt.6.0) then
        g0=6.0d0
        g1=0.0d0
c    write (2,*) 'U theta overflow - positive'
      else if (g0.lt.3.0) then
        g0=3.0d0
        g1=0.0d0
c    write (2,*) 'U theta underflow - positive'
      end if
c
      go to 98
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
c
c    KINETIC EXPRESSIONS FOR THE POSITIVE ELECTRODE
c
c    h0 is the exchange current density (A/m2)
c    h1 is the derivative of io wrt solid concentration, xx(3,j)
c    h2 is the derivative of io wrt electrolyte concen., xx(1,j)
c
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c
c    NONAQUEOUS LIQUIDS
c
  98   if (lag.eq.1) go to 99
      alpha=0.5d0
      alphc=0.5d0
      h0=rka3*dsqrt(xx(1,j))*dsqrt(ct3-xx(3,j))*dsqrt(xx(3,j))
      h1=-rka3*dsqrt(xx(1,j))*dsqrt(ct3-xx(3,j))*dsqrt(xx(3,j))*(1.0
     &  /(ct3-xx(3,j))-1.0d0/xx(3,j))/2.0d0
      h2=rka3*dsqrt(ct3-xx(3,j))*dsqrt(xx(3,j))/dsqrt(xx(1,j))/2.0d0
c
```

```
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c
c    POLYMER
c
c    alpha=0.5d0
c    alphc=0.5d0
c    h0=rka3*dsqrt(xx(1,j))*dsqrt(cmax-xx(1,j))*dsqrt(ct3-xx(3,j))
c    1*dsqrt(xx(3,j))
c    h1=-rka3*dsqrt(xx(1,j))*dsqrt(cmax-xx(1,j))*dsqrt(ct3-xx(3,j))
c    1*dsqrt(xx(3,j))*(1.0d0/(ct3-xx(3,j))-1.0d0/xx(3,j))/2.0d0
c    h2=-rka3*dsqrt(xx(3,j))*dsqrt(ct3-xx(3,j))*dsqrt(cmax-xx(1,j))
c    1*dsqrt(xx(1,j))*(1.0d0/(cmax-xx(1,j))-1.0d0/xx(1,j))/2.0d0
c
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c
    end if
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%
    if (lag.eq.1) go to 99
c
    r1=alpha*frt
    if(j.le.n1+1) then
      an1=1.0d0
      an2=0.0d0
    else
an1=0.0d0
      an2=1.0d0
    endif

    r2=r1*(xx(6,j)-xx(2,j)-g0-fc*xx(5,j)*(an1*ranode+an2*rcathde))
c
    de=-2.d0*r2-r2**3/3.d0
    if(dabs(r2).gt.200.d0) then
      if(r2.gt.200.d0) de=7.d86
      if(r2.lt.-200.d0) de=-7.d86
      pe=7.d86
    else
      if(dabs(r2).gt.1.0d-7) de=expf(-r2)-expf(r2)
      pe=expf(-r2)+expf(r2)
    endif
    b(3,1)=h2*de
    b(3,2)=h0*r1*pe
    b(3,6)=-b(3,2)
    b(3,3)=h1*de+h0*r1*g1*pe
    b(3,5)=1.0d0+fc*b(3,2)*(an1*ranode+an2*rcathde)
    g(3)=-h0*de-xx(5,j)
```

```
c
99 return
      end
c***********************************************************************
      subroutine prop(nj,n2,n1)
      implicit real*8(a-h,o-z)
      parameter(maxt=900)
      common/const/ fc,r,t,frt,cur,ep3,ep2,pi,nneg,nprop,npos,
     &    ep1,epf3,epf1,epp1,epp2,epp3,shape3,shape1,ep(221),
     &    epp(221),epf(221)
      common/var/ xp(10),xx(6,221),xi(6,221),xt(6,221,maxt)
      common/tprop/df(221),cd(221),tm(221),
     &    ddf(221),dcd(221),dtm(221),dfu(221),d2fu(221)
      common/temp/ thk,htc,dudt,Cp,dens,tam,g0,ncell,lht,qq
c
      do 99 j=1,nj
        ee=ep(j)+epp(j)
        go to (1,2,3,4,5,6,7,8,9,10,11),nprop
    1   go to 101          ! AsF6 in methyl acetate
    2   go to 102          ! Perchlorate in PEO
    3   go to 103          ! Sodium Triflate in PEO
    4   go to 104          ! LiPF6 in PC (Sony cell simulation)
    5   go to 105          ! Perchlorate in PC (West's simulation)
6   go to 106          ! Triflate in PEO
    7   go to 107          ! LiPF6 in EC/DMC (Bellcore)
    8   go to 108          ! LiPF6 in EC/DMC (Bellcore) cell #2
    9   go to 109          !Ideal polymer, t+ = 1.0
   10   go to 110          !LiTFSI in PEMO (from Steve Sloop, 1999)
   11   go to 111          ! LiPF6 in EC:DMC
c
c     &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c     AsF6 in methyl acetate
c
c     diffusion coefficient of the salt (m2/s)
  101   df(j)=(ee**1.5d0)*1.54d-09
        ddf(j)=0.0d0
c     conductivity of the salt (S/m)
        cd(j)=2.5d0*(ee**(1.5d0))
        dcd(j)=0.0d0
c     transference number of lithium
        tm(j)=0.20d0
        dtm(j)=0.0d0
c     activity factor for the salt
        dfu(j)=0.0d0
        d2fu(j)=0.0d0
go to 99
```

```
c   &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c   Perchlorate in PEO
c
c   diffusion coefficient of the salt (m2/s)
 102  df(j)=(ee**1.5d0)*1.78d-12
      ddf(j)=0.0d0
c   conductivity of the salt (S/m)
      cd(j)=1.6d-02*ee**(1.5d0)
      dcd(j)=0.0d0
c   transference number of lithium
      tm(j)=0.10d0
      dtm(j)=0.0d0
c   activity factor for the salt
      dfu(j)=0.0d0
      d2fu(j)=0.0d0
      go to 99
c   &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c   Sodium Triflate in PEO
c
 103  r0=1.3041d-07
      r1=4.4978d-07
r2=-3.1248d-07
      r3=-2.2383d-07
      r4=8.9264d-09
c
c   diffusion coefficient of the salt (m2/s)
      df(j)=0.0001d0*(ee**1.5d0)*(r0+r1*xx(1,j)/1000+
     &     r2*((xx(1,j)/1000)**0.5d0) + r3*((xx(1,j)/1000)**1.5d0)
     &     + r4*((xx(1,j)/1000)**3.0d0))
      ddf(j)=0.0001d0*(ee**1.5d0)*(r1/1000d0 +
     &     0.5d0*r2*(xx(1,j)**(-0.5d0))/(1000.0d0**0.5d0)
     &     + 1.5d0*r3*(xx(1,j)**0.5d0)/1000d0**1.5d0 +
     &     3.0d0*r4*(xx(1,j)**2.0d0)/1000d0**3.0d0)
      if (xx(1,j).ge.3.0d03) then
        df(j)=(ee**1.5d0)*1.6477d-12
        ddf(j)=0.0d0
      end if
c   conductivity of the salt (S/m)
      r7=4.32d-05
      r8=0.00017d0
      r9=0.000153d0
      r10=3.73d-05
c
cd(j)=100*(ee**(1.5d0))*(r7+r8*xx(1,j)/1000+r9*xx(1,j)*xx(1,j)
     &      /1000000+r10*xx(1,j)*xx(1,j)*xx(1,j)/1000000000)
      dcd(j)=100*(ee**(1.5d0))*(r8/1000+2.0*r9*xx(1,j)/1000000+3.0*r10
    &      *xx(1,j)*xx(1,j)/1000000000)
c
c     transference number of lithium
c
       if(xx(1,j).lt.0.3d03) then
         r5=0.32141d0
         r6=2.5768d0
         r11=71.369d0
         r12=643.63d0
         r13=1983.7d0
         r14=2008.d0
         r15=287.46d0
         tm(j)=r5-r6*xx(1,j)/1000.+r11*xx(1,j)*xx(1,j)/1000000.
    &     -r12*((xx(1,j)/1000.)**(3.0d0))+r13*((xx(1,j)/1000.)**4.0d0)
    &     -r14*((xx(1,j)/1000.)**(5.0d0))+r15*((xx(1,j)/1000.)**6.0d0)
         dtm(j)=-r6/1000.+2.0d0*r11*xx(1,j)/1000000.-
    &          3.0d0*r12*(xx(1,j)**2.0d0)/(1000.**3.0d0) +
    &          4.0d0*r13*(xx(1,j)**3.0d0)/(1000.**4.0d0) -
    &          5.0d0*r14*(xx(1,j)**4.0d0)/(1000.**5.0d0) +
&          6.0d0*r15*(xx(1,j)**5.0d0)/(1000.**6.0d0)
       else
         tm(j)=0.0d0
         dtm(j)=0.0d0
       end if
c
       if(xx(1,j).ge.0.70d03) then
         r5=4.5679d0
         r6=4.506d0
         r11=0.60173d0
         r12=1.0698d0
         tm(j)=-r5+r6*expf(-((xx(1,j)/1000.-r11)/r12)**2.)
         dtm(j)=-r6*(xx(1,j)/1000.-r11)*2.
    &          *expf(-((xx(1,j)/1000.-r11)/r12)**2.)/r12/r12/1000.
       end if
c
       if(xx(1,j).ge.2.58d03) then
         tm(j)=-4.4204d0
         dtm(j)=0.0d0
       end if
c
c     activity factor for the salt:  (dlnf/dc) and (d2lnf/dc2)
c
       if(xx(1,j).gt.0.45d03) then
         r17=0.98249d0
         r18=1.3527d0
         r19=0.71498d0
         r20=0.16715d0
         r21=0.014511d0
         thermf=r17-r18*xx(1,j)/1000.+r19*xx(1,j)*xx(1,j)/1000000.-
     &    r20*xx(1,j)*xx(1,j)*xx(1,j)/1000000000.+r21*xx(1,j)*xx(1,j)
     &    *xx(1,j)*xx(1,j)/1000000000000.
         dthermf=-r18/1000.+2.*r19*xx(1,j)/1000000.-
     &    3.*r20*xx(1,j)*xx(1,j)/1000000000.+4.*r21*xx(1,j)*xx(1,j)
     &    *xx(1,j)/1000000000000.
       end if
c
       if(xx(1,j).le.0.45d03) then
         r23=0.99161d0
         r24=0.17804d0
         r25=55.653d0
         r26=303.57d0
         r27=590.97d0
         r28=400.21d0
thermf=r23-r24*xx(1,j)/1000.-r25*xx(1,j)*xx(1,j)/1000000.+
     &    r26*xx(1,j)*xx(1,j)*xx(1,j)/1000000000.-r27*xx(1,j)*xx(1,j)
     &    *xx(1,j)*xx(1,j)/1000000000000.+r28*xx(1,j)*xx(1,j)*xx(1,j)
     &    *xx(1,j)*xx(1,j)/1000000000000000.
         dthermf=-r24/1000.-2.*r25*xx(1,j)/1000000.+
     &        3.*r26*xx(1,j)*xx(1,j)/1000000000.-4.*r27*xx(1,j)
     &        *xx(1,j)*xx(1,j)/1000000000000.+5.*r28*xx(1,j)*xx(1,j)
     &        *xx(1,j)*xx(1,j)/1000000000000000.
       end if
c
       dfu(j)=(-1.+2.*thermf)/xx(1,j)
       d2fu(j)=1./xx(1,j)/xx(1,j)-2.*thermf/xx(1,j)/xx(1,j)+
     &      2.*dthermf/xx(1,j)
c
       if(xx(1,j).ge.3.00d03) then
         dfu(j)=-0.9520/xx(1,j)
         d2fu(j)=0.9520/xx(1,j)/xx(1,j)
       end if
       go to 99
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    LiPF6 in PC (Sony cell simulation)
c
c    this is actually the diff coeff for perchlorate
c    diffusion coefficient of the salt (m2/s)
 104   df(j)=(ee**1.5d0)*2.58d-10
       ddf(j)=0.0d0
c    conductivity of the salt (S/m)
c    pmax=0.5409
       pmax=0.035d0
```

```
      pu=0.857d0
      aa=1.093
      bb=0.04d0
      rho=1.2041d03
      fun=pmax*((1.0d0/rho/pu)**aa)*expf(bb*((xx(1,j)/rho-pu)**2.0)
     &      -(aa/pu)*(xx(1,j)/rho-pu))
      fun2=2.0d0*(bb/rho)*(xx(1,j)/rho-pu)-aa/pu/rho
      cd(j)=0.0001+(ee**1.5d0)*((xx(1,j))**aa)*fun
      dcd(j)=(ee**1.5d0)*fun*(aa*(xx(1,j)**(aa-1.0d0))+(xx(1,j)**aa)
     &      *fun2)
c    transference number of lithium
      tm(j)=0.20d0
      dtm(j)=0.0d0
c
c    activity factor for the salt (dlnf/dc and d2lnf/dc2)
      dfu(j)=0.0d0
      d2fu(j)=0.0d0
      go to 99
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    Perchlorate in PC (West's simulation)
c
c    diffusion coefficient of the salt (m2/s)
 105   df(j)=(ee**1.5d0)*2.58d-10
      ddf(j)=0.0d0
c    conductivity of the salt (S/m)
      pmax=0.542d0
      pu=0.6616d0
      aa=0.855d0
      bb=-0.08d0
      rho=1.2041d03
      fun=pmax*((1.0d0/rho/pu)**aa)*expf(bb*((xx(1,j)/rho-pu)**2.0)
     &      -(aa/pu)*(xx(1,j)/rho-pu))
      fun2=2.0d0*(bb/rho)*(xx(1,j)/rho-pu)-aa/pu/rho
      cd(j)=0.0001+(ee**1.5d0)*((xx(1,j))**aa)*fun
      dcd(j)=(ee**1.5d0)*fun*(aa*(xx(1,j)**(aa-1.0d0))+(xx(1,j)**aa)
     &      *fun2)
c    transference number of lithium
   tm(j)=0.20d0
      dtm(j)=0.0d0
c    activity factor for the salt
      dfu(j)=0.0d0
      d2fu(j)=0.0d0
      go to 99
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    Triflate in PEO
c
```

```
 106  r0=-5.0891863844d-05
      r1=8.38645199394d-07
      r2=-5.19747901855d-10
      r3=8.0832709407d-14
c    diffusion coefficient of the salt (m2/s)
      df(j)=(ee**1.5d0)*7.5d-12
      ddf(j)=0.0d0
c    conductivity of the salt (S/m)
      cd(j)=(ee**(1.5d0))*100.0d0*(r0 + r1*xx(1,j)
     &       +r2*xx(1,j)*xx(1,j)+r3*xx(1,j)*xx(1,j)*xx(1,j))
      dcd(j)=(ee**1.5d0)*100.0d0*(r1 + 2.0d0*r2*xx(1,j) +
     &       3.0d0*r3*xx(1,j)**2.0d0)
c    transference number of lithium
c    rough conc. dependence of t+ - highly suspect
      tm(j)=0.0107907d0 + 1.48837d-04*xx(1,j)
      dtm(j)=1.48837d-04
c    activity factor for the salt
      dfu(j)=0.0d0
      d2fu(j)=0.0d0
      go to 99
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    LiPF6 in EC/DMC and p(VdF-HFP) (Bellcore)
c    This is the 1:2 v/v mixture of EC/DMC (eq. 2 of paper)
c    D and t+ given below were fit from discharge curves
c
c    diffusion coefficient of the salt (m2/s)
c    107 df(j)=(ee**1.5d0)*9.00d-11
 107  df(j)=(ee**3.3d0)*7.50d-11
      ddf(j)=0.0d0
c
c    conductivity of the salt (S/m)
c    This is the conductivity of the liquid + salt only (no polymer)
c
c    kappa (c) for EC/DMC 2:1 with LiPF6 at 25 deg C
c    cd(j)=(ee**1.5d0)*(0.0911+1.9101*xx(1,j)/1000-1.052*xx(1,j)*
c    1xx(1,j)/1000/1000+0.1554*(xx(1,j)/1000)**3.0d0)
c    derivative of kappa (c) for EC/DMC 2:1 at 25 deg C
c    dcd(j)=(ee**1.5d0)*(1.9101/1000-2.0*1.052*xx(1,j)/1000/1000
c    1+3.0*0.1554/1000*(xx(1,j)/1000)**2.0d0)
c
c    kappa (c) for EC/DMC 1:2 w/ LiPF6 at 25 deg C
c    Note Bruggeman exponent should be adjusted to account for
c    polymer phase - this also affects "fac" parameter in Ohm's
c    law equation number 2
      r1=0.00010793d0
      r2=0.0067461d0
      r3=0.0052245d0
      r4=0.0013605d0
      r5=0.00011724d0
      cd(j)=(ee**3.3d0)*(r1+r2*xx(1,j)/1000.d0
c     cd(j)=(ee**1.5d0)*(r1+r2*xx(1,j)/1000.d0
     &   -r3*xx(1,j)*xx(1,j)/1000000.d0
     &   +r4*(xx(1,j)/1000.d0)**3.0d0-r5*(xx(1,j)/1000.d0)**4.0d0)*100.d0
      dcd(j)=(ee**3.3d0)*(r2/10-r3*2.0d0*xx(1,j)/10000.d0
c     dcd(j)=(ee**1.5d0)*(r2/10-r3*2.0d0*xx(1,j)/10000.d0
     &       +3.0d0*r4*xx(1,j)*xx(1,j)/10000000.d0
     &       -0.4d0*r5*(xx(1,j)/1000.d0)**3.0d0)
c
c     transference number of lithium
      tm(j)=0.363d0
      dtm(j)=0.0d0
c
c     activity factor for the salt (dlnf/dc and d2lnf/dc2)
      dfu(j)=0.0d0
      d2fu(j)=0.0d0
      go to 99
c     &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c     LiPF6 in EC/DMC and p(VdF-HFP) (Bellcore) cell #2
c     This is the 2:1 v/v mixture of EC/DMC (eq. 1 of paper)
c     D and t+ given below were fit from discharge curves
c
c     diffusion coefficient of the salt (m2/s)
c     108 df(j)=(ee**1.5d0)*9.00d-11
  108 df(j)=(ee**3.3d0)*7.50d-11
      ddf(j)=0.0d0
c
c     conductivity of the salt (S/m)
c
c     kappa (c) for EC/DMC 2:1 w/ LiPF6 at 25 deg C
c     Note Bruggeman exponent should be adjusted to account for
c     polymer phase - this also affects "fac" parameter in Ohm's
c     law equation number 2
      r1=0.00041253d0
      r2=0.005007d0
      r3=0.0047212d0
      r4=0.0015094d0
c     r5=0.0016018d0
      r5=0.00016018d0
      cd(j)=(ee**3.3d0)*(r1+r2*xx(1,j)/1000.d0
c     cd(j)=(ee**1.5d0)*(r1+r2*xx(1,j)/1000.d0
     &       -r3*xx(1,j)*xx(1,j)/1000000.d0
     & +r4*(xx(1,j)/1000.d0)**3.0d0-r5*(xx(1,j)/1000.d0)**4.0d0)*100.d0
```

```
      dcd(j)=(ee**3.3d0)*(r2/10-r3*2.0d0*xx(1,j)/10000.d0
c    dcd(j)=(ee**1.5d0)*(r2/10-r3*2.0d0*xx(1,j)/10000.d0
     &      +3.0d0*r4*xx(1,j)*xx(1,j)/10000000.d0
     &      -0.4d0*r5*(xx(1,j)/1000.d0)**3.0d0)
c
c    transference number of lithium
      tm(j)=0.363d0
      dtm(j)=0.0d0
c
c    activity factor for the salt (dlnf/dc and d2lnf/dc2)
      dfu(j)=0.0d0
      d2fu(j)=0.0d0
      go to 99
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    Ion Exchange Membrane, t+ = 1.0
 109  continue
      df(j) = (ee**1.5)*1.0d-11
      ddf(j) = 0.0d0
      cd(j) = (ee**1.5)*0.01d0
      dcd(j) = 0.0
      tm(j) = 1.0d0
      dtm(j) = 0.0d0
      dfu(j) = 0.0d0
      d2fu(j) = 0.0d0
      go to 99
c
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    LiTFSI in PEMO from Steve Sloop and John Kerr
 110  continue
      df(j) = (ee**1.5)*((-3.0d-17)*xx(1,j) + 6.0d-13)
      ddf(j) = (ee**1.5)*(-3.0d-17)
cd(j) = (ee**1.5)*expf((-3.0d-07)*xx(1,j)*xx(1,j)
     &      + 6.5d-04*xx(1,j) - 1.12)
      dcd(j) = cd(j)*((-6.0d-07)*xx(1,j) + 6.5d-04)
      tm(j) = 1.6*expf(-0.0017*xx(1,j))
      dtm(j) = tm(j)*(-0.0017)
      dfu(j) = 0.0d0
      d2fu(j) = 0.0d0
      go to 99
c
c    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    LiPF6 in EC:DMC.  D is for LiAsF6 in methyl formate,
c    corrected for viscosity by Walden's Rule, from the
c    Organic Electrolyte Handbook (Janz).  t+ is for LiClO4
c    in various electrolytes.  cd is from measurements made at
c    Bellcore, as reported in Marc Doyle's dissertation.
```

```
111   continue
      df(j) = (ee**1.5)*1.7d-10
      ddf(j) = 0.0d0
      cd(j) = (ee**1.5)*(0.0911+1.9101*xx(1,j)/1000.0 -
    &       1.052*((xx(1,j)/1000.0)**2.0) +
    &       0.1554*((xx(1,j)/1000.0)**3.0))
      dcd(j) = (ee**1.5)*(1.9101/1000.0 –
    &       2.0*1.052*xx(1,j)/1000.0/1000.0
    &       + 0.1554*3.0*((xx(1,j)/1000.0)**2.0)/1000.0)
      tm(j) =0.3d0
      dtm(j) = 0.0d0
      dfu(j) = 0.0d0
      d2fu(j) = 0.0d0
c     &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
   99 continue
c
      return
      end
c*******************************************************************
c     That's All Folks!
c*******************************************************************
```

## 6. Bibliography

1. R. Zallen, *The Physics of Amorphous Solids*, p 135, John Wiley & Sons, New York, (1983)
2. C. Rajagopal and M. Satyam, *J. Appl. Phys.*, **49**, 5536 (1978)
3. M. B. Heaney, *Physica A*, **241**, 296 (1997)
4. F. Carmona, P. Prudhon and F. Barreu, *Solid State Communications*, **51**, 255 (1984)
5. Z. Rubin, A. Sunshine, M.B. Heaney, I. Bloom and I. Balberg, *Phys. Rev. B*, **59**, 196 (1999)
6. P. F. Carcia, A. Ferretti and A. Suna, *J Appl Phys.*, **53**, 5282 (1982)
7. R. W. A. Franco, J. P. Donoso, C. J. Magon, A. O. Florentino, M. J. Saeki, J. M. Pernaut and A. L. de Oliveira, *Solid State Ionics*, **113-115**, 149 (1998)
8. G. E. Pike and C. H. Seager, *J Appl Phys.*, **48**, 5152 (1977)
9. F. A. Modine, A. R. Duggal, D. N. Robinson, E. L. Churnetski, M. Bartkowiak, G. D.Mahan and L. M. Levinson, *J Mater Res.*, **11**, 2889 (1996)
10. E. K. Sichel, J. I. Gittleman and P. Sheng, *Phys Rev B*, **18** 5712, (1978)
11. Y.-M. Chiang, L. A. Silverman, R. H. French and R. M. Cannon, *J Am Ceram Soc.*, **77** 1143 (1994)
12. J. Israelachvili, *Intermolecular & Surface Forces*, Academic Press, London (1992).
13. J. S. Reed, *Principles of Ceramics Processing*, John Wiley & Sons, New York (1995)
14. H. Tukamoto and A. R. West, "Electronic Conductivity of $LiCoO_2$ and Its Enhancement by Magnesium Doping" *J. Electrochem. Soc.,* **144** 3164 (1997)
15. P.P. Soo, B Huang, D. R. Sadoway and A. M. Mayes, "Rubbery Block Copolymer Electrolytes for Solid-State Rechargeable Lithium Batteries" *J. Electrochem. Soc.,* **146** 32 (1999)
16. J. Newman, *Electrochemical Systems*, Prentice Hall, NJ, 1991
17. A.J. Vaccaro, T. Palanisamy, R.L. Kerr and J.T. Maloy, *J. Electrochem. Soc.*, **129**, 682 (1982)
18. D. Guyomard and J.M. Tarascon, "Li Metal-Free Rechargeable $LiMn_2O_4$/Carbon Cells: Their Understanding and Optimization" *J. Electrochem. Soc.*, **139**, 937 (1992)
19. Y-M. Choi and S-I. Pyun, "Determination of electrochemical active area of porous $Li_{1-\delta}CoO_2$ electrode using the GITT technique" *Solid State Ionics*, **109**, 159 (1998)
20. R.E. Meredith and C.W. Tobias in P. Delahay and C.W. Tobias, ed. *Advances in Electrochemistry and Electrochemical Engineering*, p. 26. John Wiley & Sons, NY, 1962
21. M. Doyle, T.F. Fuller and J. Newman, "Modeling of Galvanostatic Charge and Discharge of the Lithium/Polymer/Insertion Cell", *J. Electrochem. Soc.*, **140**, 1526-1533 (1993)
22. M. Doyle, "Design and Simulation of Lithium Rechargeable Batteries", *Ph.D. Thesis,* **University of California, Berkeley**, (1995)

23. C.R. Pals and J. Newman, "Thermal Modeling of the Lithium/Polymer Battery I.", *J. Electrochem. Soc.*, **142**, 3274-3281 (1995)
24. C.R. Pals and J. Newman, "Thermal Modeling of the Lithium Polymer Battert II.", *J. Electrochem Soc.*, **142**, 3282-3288 (1995)
25. R. Darling and J. Newman, "Modeling a Porous Intercalation Electrode with Two Characteristic Particle Sizes", *J. Electrochem. Soc.*, **144**, 4201-4208 (1997)
26. I.J. Ong and J. Newman, "Double-Layer Capapcitance in a Dual Lithium Ion Insertion Cell", *J. Electrochem. Soc.*, **146**, 4360-4365 (1999)
27. R. Darling and J. Newman, "Modeling Side Reactions in Composite $Li_yMn_2O_4$ Electrodes", *J. Electrochem. Soc.*, **145**, 990-998 (1998)
28. T. F. Fuller, M. Doyle and J. Newman, "Simulation and Optimization of the Dual Lithium Ion Insertion Cell", *J. Electrochem. Soc.*, **141**, 1-10 (1994)
29. M. Doyle, J. Newman, "Analysis of capacity-rate data for lithium batteries using simplified models of the discharge process", *J. Appl. Electrochem.*, **27**, 846-856, (1997)
30. G.S. Nagarajan, J. W. Van Zee and R.M. Spotnitz, "A Mathematical Model for Intercalation Electrode Behavior I.", *J. Electrochem. Soc.*, **145**, 771-779 (1998)
31. J. Newman and C.W. Tobias, "Theoretical Analysis of Current Distribution in Porous Electrodes", *J. Electrochem. Soc.*, **109**, 1183 (1962)

## 7. Biographical Note

The author was born in Goettingen, Germany in 1974. He attended Miramonte High School in Orinda, CA after moving there in 1986. After graduating in 1993, he came to MIT to pursue a Bachelor's and later a Master's degree in Materials Science and Engineering. After completing the degree, he is moving to New York City to take an analyst position with Goldman Sachs.