UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

FINAL DEGREE PROJECT

Degree in Chemical Engineering

ELECTRO-THERMAL MODELING OF ELECTROCHEMICAL BATTERIES

REPORT

Author: Pol Miró Jané
Supervisor: Dr. Francesc Estrany Coda
Co-supervisor: Dr. Àngel Cuadras Tomàs

Call: Autumn 2016-2017

A thesis submitted in partial fulfilment of the requirements
for the degree of Chemical Engineering

EEBE-UPC

Resum

En l'època de l'electrònica, les bateries electroquímiques guanyen importància per facilitar la mobilitat d'ordenadors, telèfons, rellotges, etc. Més enllà de l'electrònica, podem veure com, fins i tot, els vehicles elèctrics requeriran una gran quantitat de bateries. Actualment, les bateries recarregables més empleades són les de metall- hidrur i liti. Encara que hi ha models matemàtics per descriure aquest últim tipus de bateria, poc s'ha treballat sobre l'envelliment.

Aquest projecte presenta un model termo-elèctric d'una bateria genèrica de metall- hidrur, mitjançant tècniques numèriques d'elements finits, implementades en Matlab®, reproduint els resultats disponibles en la literatura.

Tot seguit, s'estudia l'envelliment a partir de l'entropia irreversible modificant paràmetres crítics, com la porositat o la concentració de l'espècie en la fase sòlida. S'analitzen cicles de càrrega i descàrrega, concloent que la principal contribució a l'envelliment de la bateria es la pèrdua de capacitat dels elèctrodes.

Resumen

En el apogeo de la electrónica las baterías electroquímicas ganan importancia para facilitar la movilidad de ordenadores, teléfonos, relojes, etc. Más allá de la electrónica, los vehículos eléctricos requerirán gran cantidad de baterías. Actualmente, las baterías recargables más usadas son las de metal-hidruro y las de litio. Aunque hay modelos matemáticos para describir este último tipo de baterías, poco se ha trabajado sobre el envejecimiento.

Este proyecto presenta un modelo termo-eléctrico de una batería genérica de metal-hidruro empleando técnicas numéricas de elementos finitos implementadas en Matlab®, reproduciendo los resultados disponibles en la literatura.

Después, se estudia el envejecimiento a partir de la entropía irreversible cambiando parámetros críticos, como la porosidad o la concentración de la especie que se cicla. Se analizan ciclos de carga y descarga, concluyendo que la pérdida de capacidad de los electrodos es la principal contribución al envejecimiento.

Abstract

In the apogee of electronics, electrochemical batteries gain importance, not only in electrical vehicles but also in consumer electronic applications. Alongside with lithium (Li-ion) batteries, nickel/metal hydride batteries (Ni-MH) are the most widely used. While there are mathematical models available in literature, little is known of its aging.

This work presents an electro-thermal model of a generic nickel/metal hydride battery using CFD technics in Matlab® achieving suitable accordance with early researches. Aging is then assessed by means of irreversible entropy and by changing key parameters such as porosity and concentration of cyclable species. Charge and discharge cycles are conducted, concluding that the electrode's loss of capacity is the largest contribution to the battery's aging.

Acknowledgments

I would like to thank my advisors Dr. Àngel Cuadras and Dr. Francesc Estrany for their support, patience and corrections throughout the project. Their knowledge, skills and corrections have been most valued in this work.

Many thanks to my brother Arnau Miró who has introduced me into the CFD and for providing useful tools in the realization of this project. I am thankful for my parents whose efforts have made my academic life possible.

I am also grateful to my friends Julio Uribe and Cesar Zapata for their support, motivation and for sharing their insights on this project.

My most sincere gratitude to professor John Newman (Department of Chemical Engineering, University of California) for responding my emails and providing priceless understanding on this project.

Last and not least, my most sincere thanks to my girlfriend Andrea Poveda for her outstanding support and patience. She put up with my talk and despair tirelessly, she earned heaven for that. She also helped me keep my horizons broad while working on this project.

Motivation

The growing use of electronic devices powered by batteries is undeniable. Energy storage applications can be scaled from small applications such as smartphones or larger like HEVs (hybrid electric vehicles), i.e., batteries used as replacement or complement of liquid fuels, (Rahn and Wang 2012).

Therefore, the need to know the remaining battery charge, known as state of charge (SOC), emerged. Many authors have investigated the issue and mathematical models describing the behaviour of electrochemical batteries have been developed. A reference in this area is professor Newman, who has enormously contributed in the modelling of electrochemical systems (Bernardi 1985; Fuller, Thomas F. Doyle, Marc. Newman 1994a; Fuller, Thomas F. Doyle, Marc. Newman 1994b; J Newman 2015; John Newman 1968; John Newman et al. 2003; Pals and Newman 1995; B. Paxton and Newman 1997; Rao 1997).

The field of battery systems is always on the move, looking for a better battery, e.g., higher energy, power, greater safety, lower cost. It is reasonable to assume that practical research will split into two branches, one focused on developing new materials, i.e., new types of batteries such as lithium polymer and the other will try to increase the understanding on battery systems, e.g., real time SOC prediction, degradation processes or side reactions.

Battery systems engineering requires a multidisciplinary approach. Chemists and chemical engineers understand the electrochemistry involved in batteries, but they may lack the background to assess complex mathematical modelling and computer algorithms required for modelling a battery. Mathematical modellers or other engineers may be able to develop accurate models of batteries, but these are seldom easily adopted for systems engineering owing to the complexity of partial differential equations. Systems engineers have the background to analyse and simulate the system response but may not understand the underlying chemistry or modelling, (Rahn and Wang 2012).

For that, there is software that efficiently solve the set of differential equations that comprise a battery, nonetheless, these are in the end black tool boxes. Only by undergoing the path of learning and implementing algorithms involved in solving the problem itself one can truly learn, albeit these programmes are useful to verify solutions. Thus, the importance of knowing the procedure involved in problem solving to avoid unrealistic, however mathematical correct, solutions.

SYMBOLS

Symbol	Name	Units
ε	Porosity or tolerance	-
D	Diffusion coefficient	cm^2/s
k	Conductivity of the electrolyte or relaxation factor	$\frac{S}{cm}\ or -$
F	Faraday's constant	C/mol
R	Universal gas constant	$J/molK$
T	Temperature	K
η	Overpotential	V
a	Specific interfacial area	cm^2/cm^3
i^o	Exchange current density	A/cm^2
i	Current density	A/cm^2
j_r	Reaction current density due to production of consumption of a species	A/cm^3
α_a, α_c	Transfer coefficients	-
c	Volume-averaged concentration of a species over a phase	mol/cm^3
ρ	Density	g/cm^3
M_{KOH}	Molecular weight of KOH	g/mol
M_{H_2O}	Molecular weight of water	g/mol
t^-	Transport number	-
r	radial coordinate	cm
x	x coordinate	cm
σ	Electrical conductivity	S/cm
ϕ	Potential	V

$f^{\mp}$	Mean molar coefficient of KOH	-
I_{cell}	Applied current density	A/cm^2
V_{cell}	Cell potential	V
E_{act}	Activation energy	J/mol
U	Open circuit potential	V
q	Volumetric heat generation rate	$J/cm^3 s$
L_{cell}	Cell width	cm
L_{MH}	MH electrode thickness	cm
L_{Ni}	Nickel electrode thickness	cm
L_{sep}	Separator thickness	cm
h	Equivalent heat transfer coefficient	$W/cm^2 K$
λ	Thermal conductivity	$W/cm\ K$
C_p	Specific heat	J/gK
$\dot{S}_l$	Local entropy rate	$W/cm^3 K$
$\dot{S}$	Entropy rate	$W/cm^2 K$
t	Time	s
Q	Maximum charge per unit of projected area of the electrode	C/cm^2
DOD	Depth of discharge	-
S	Source term	-
S_c	Constant part of source term	-
S_p	Linear part of source term	-

Subscripts

s	Solid phase or separator
e	Liquid phase
o	Water in the electrolytic solution

ref	Reference
MH, Ni, Sep	Referring to nickel electrode, MH electrode or separator
se	Solid/electrolyte interface

Superscript

eff	Effective

Other symbols

p	Property of a material (Chapter 2)
Γ	Property pertinent to a variable (Chapter 2)
k^o	Standard rate constant
k_f, k_b	Rate constant for a forward and backward process (Chapter 1)
C_f, C_b	Concentration for a forward and backward process (Chapter 1)
v_{net}	Net rate for a reaction (Chapter 1)
ϕ	General variable or property

CONTENTS

List of figures

INTRODUCTION

This work consists on a mathematical model of a generic metal hydride battery and a preliminary assessment of its aging via entropy. The objective is to achieve proper accordance with already developed models (B. Paxton and Newman 1997). Nevertheless, little is known about the degradation (or aging) of batteries. The current literature focuses on degradation mechanisms (Chehab et al. 2006; Serrao et al. 2005; Young and Yasuoka 2016; Hu et al. 2016; Albertus, Christensen, and Newman 2008). Recent studies show that according to the second and third laws of thermodynamics, entropy can be used as a measure to describe battery degradation (Cuadras, Ovejas, and Quilez 2013; Bryant 2014).

Though lithium-ion batteries offer better characteristics than those of metal hydride, they also present a higher challenge as far as modelling is concerned. Metal hydrides, such as $La_{1/6}Ni_{5/6}$, are certain compounds that used as electrodes provide a less toxic alternative to cadmium electrodes in nickel/cadmium battery systems (B. K. Paxton 1995).

A battery system, regardless of its kind, is composed of a certain governing PDEs (partial differential equation), these must be discretised in space to reduce them to ODEs (ordinary differential equations). Numerical methods must be employed to discretize the equations, in this work a finite volume method following Patankar's book(Patankar 1980) is used. Then, those equations are solved iteratively with the TDMA solver provided by Miró (Miró Jané 2014).

That sort of numerical approach is computational fluid dynamics (CFD), which is a well-established tool for numerical analysis of fluid flow, mass, heat transport and chemical reactions. In that way, the battery model can solve the whole battery problem from anode to cathode in terms of space and time in a single solver (W. Gu, Wang, and Liaw 1997). What is more, the algorithm is intended to be generic in order to cover a wide range of battery chemistries.

Thermal control is needed to avoid runaway and malfunctioning and even though there is much literature regarding this, no electro-thermal model has been found that provide enough parameters to make it replicable. Therefore, literature has been followed regarding thermal modelling (Bernardi 1985; Pals and Newman 1995; Rao 1997; W. B. Gu and Wang 2000a) in order to properly include the thermal effects.

Finally, the degradation of the battery is assessed via irreversible entropy. According to the second law of thermodynamics, entropy is a valid indicator to evaluate the system's decay. Entropy increases monotonously and thus a threshold of maximum wear out can be defined (Cuadras, Ovejas, and Quilez 2013).

Chapter 1 State of the art

This chapter focuses on the contributions to battery modelling made by different authors and on general theory of electrochemistry. It starts with a summary of the different models found in literature. Then, the essentials of electrochemistry needed to formulate a model are presented following (Bard and Faulkner 2001; Kiehne 2003; Rahn and Wang 2012; J Newman 2015).

1.1. Previous work

Many authors have significantly contributed to the field of modelling electrochemical batteries. This section aims to provide a general research storyline regarding metal hydride and lithium batteries over the recent decades. The storyline has been divided into three: general modelling, thermal modelling and aging analysis.

Figure 1 illustrates the major contributions to electrochemical batteries modelling. Doyle, Fuller and Newman modelled and optimised a dual lithium cell 1993 and 1995. In 1994, Timmerman and Wiedner studied the effect of proton diffusion in MH batteries. In 1995, Pauline de Vidts et al. developed a mathematical model for the discharge of a single metal hydride electrode. Two years later Paxton and Newman modelled a full MH battery, one year later Gu and Wang developed a micro-macroscopic model for MH batteries and also studied the characteristics of charge-discharge including evolution. Since then lithium batteries have been studied in a greater deal than metal-hydride, the last known publication was in 2004 when Wang and Liaw studied fast-charging metal hydride batteries.

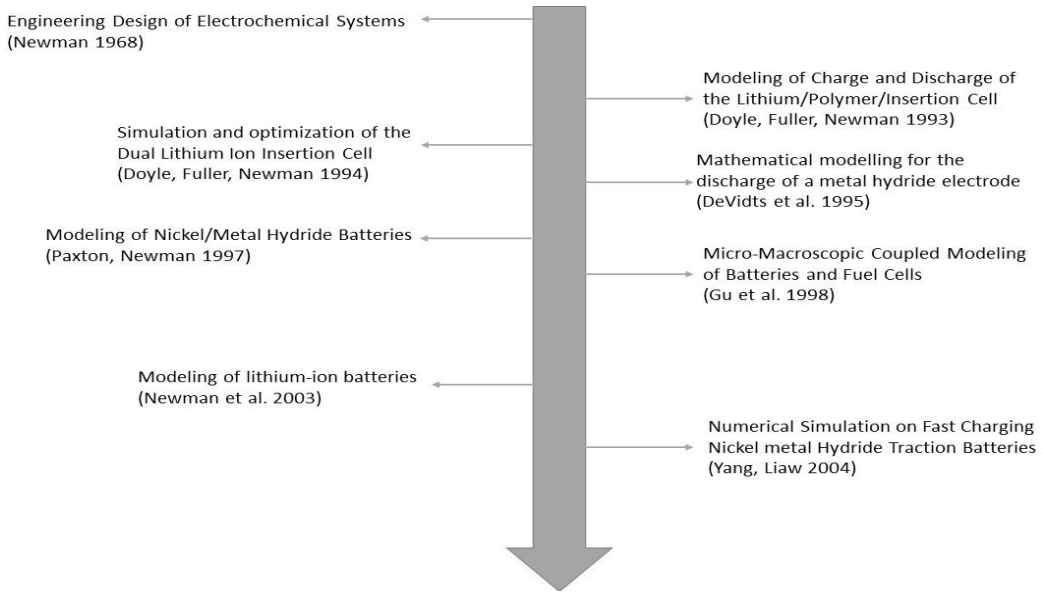

Figure 1. Major contributions to electrochemical modelling since 1968.

Figure 2 shows the contributions to thermal modelling. In 1985 Bernardi et al. developed a general energy balance for battery systems, ten year later Pals and Newman developed a thermal model for lithium batteries. In 1997 Rao and Newman developed another energy balance for insertion battery systems via enthalpy potentials. Wu et al. studied heat dissipation by attaching cold plates into MH batteries in 2000. In the same year, Gu and Wang gave a general approach of thermal-electrochemical approach of battery systems. First Yang and Liaw published numerical simulations on fast charging MH batteries in 2004 and later Wu et al. in 2006 did a thermal analysis on the same type of batteries. Thermal analysis of MH batteries during charge cycle was conducted by Jahantigh and Afshari in 2008. Finally, in 2014 Taheri et al. assessed the behaviour of MH batteries during fast charging.

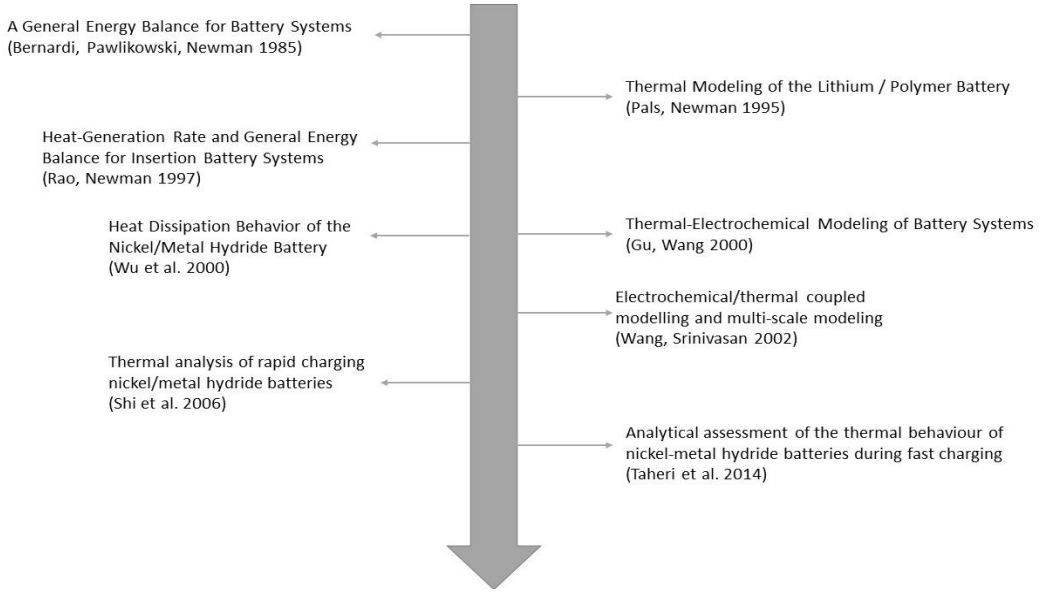

Figure 2. *Major contributions to thermal modelling since 1985.*

Little has been found regarding the degradation of a battery, figure 3 shows the major contributions in this field. Albertus, Christensen and Newman discussed and modelled the side reactions and nonisothermal effects in metal hydride batteries in 2008. Bryant in 2014 stated that entropy is a fundamental quantity to describe ageing and degradation. Cuadras et al. in 2014 evaluated the state of life of lithium batteries from loss of capacity. Young and Yasuoka did a review of capacity degradation mechanisms in MH batteries in 2016, in the same year Hu et al. studied battery health prognosis for electric vehicles using sample entropy and a predictive model.

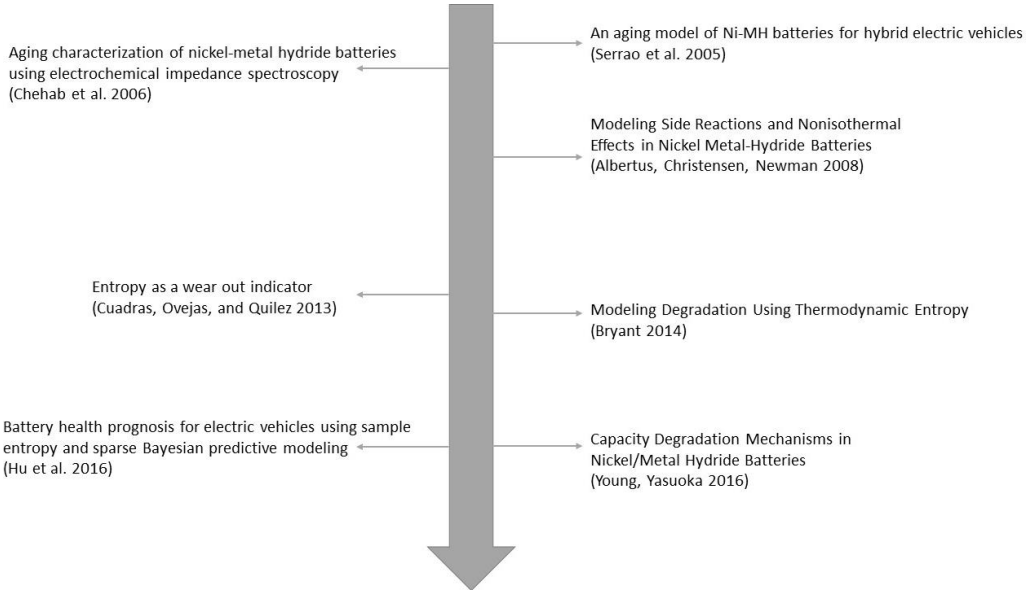

Figure 3. *Major contributions to aging.*

1.2. Electrochemical fundamentals

Every electrochemical system contains two electrodes separated by an electrolyte and connected by an external electronic conductor, as illustrated in figure 4. Ions flow through the electrolyte from one electrode to the other, and the circuit is completed by electrons flowing through the external conductor. The electrodes allow transport of electrons; the electrolyte blocks them but allow transport of ions.

It is important to differentiate between an electrochemical reaction and a redox reaction; while the second one occurs in the same place, in an electrochemical reaction reduction occurs in one electrode and oxidation in the other, albeit the rates of reactions are coupled by the principles of charge and electro-neutrality. Thus, the complete reaction is divided into two half cells and they are heterogeneous, i.e. reaction takes places at the interface between electrolyte and electrode.

Therefore, two types of cells can be found. Reaction occur spontaneously at electrodes when they are connected externally by a conductor in galvanic cells and reactions occur due to an imposition of an external voltage greater than the open-circuit voltage, which is the potential when the cell is disconnected from any circuit, in electrolytic cells. Therefore, both behaviours are given in a rechargeable battery: while it is discharged, it works as a galvanic cell and when charged as an electrolytic one. At the same time, if a battery is rechargeable is called secondary battery, for example the above-mentioned and if not, is called primary, for example Leclanché battery (Kiehne 2003).

Since first Alexander Volta invented the Volta pile in 1800, many other kinds of batteries have emerged. The most common nowadays are: lead-acid (VRLA), metal hydride (MH) and lithium ion (Li-ion).

VRLA batteries are one of the oldest rechargeable technologies with over 150 years of history. Thus, they have been widely used in both industry and consumer products, mainly because of their high reliability (Hejabi, Oweisi, and Gharib 2006). In VRLA batteries, acid is consumed in both the negative and positive to form $PbSO_4$ during discharge, and it is converted back into acid during charge.

Li-ion batteries offer the longest cycle life and the lowest self-discharge rate. They use lithium, which is the metal with lowest density and with the greatest electrochemical potential and energy-to-weight ratio. Typically, one electrode is a lithium metal oxide ($LiMO_2$) and the other a lithiated carbon (Li_xC) (Rahn and Wang 2012). The electrolyte is a liquid or gel-polymer that enable lithium ions to diffuse throughout the cell. The lithium ions insert into or deinsert from the active materials via an intercalation process.

Ni-MH battery cells have much in common with Li-ion cells: both have a unary electrolyte and use an intercalation process. The metal M in one of the electrodes is an intermetallic compound, i.e., a solid phase containing two or more metallic elements (Schulze 1974), the other electrode is nickel oxide hydroxide. That intermetallic compound is usually a mixture of lanthanum or cerium with nickel or cobalt (Kopera 2004).

Though lithium ion batteries offer better performance, they also require a more complex modelling, consequently, nickel-metal batteries have been chosen as they offer better life cycle, capacity and recharge capability over VRLA batteries. (Rahn and Wang 2012).

Recalling that a cell is divided into three parts (negative electrode, separator and positive electrode), the Ni-MH is a two-phase system consisting on a liquid phase and a solid phase. The liquid phase is the electrolyte, which is usually potassium hydroxide (KOH) that flows through the whole battery. The solid phase is the matrix of each electrode. The electrodes are porous, i.e., they are composed of active particles instead of a uniform foil or plate. These active particles are usually modelled as spheres (B. K. Paxton 1995). The separator is usually made of felt or a porous nylon material. In the Ni-MH batteries the positive electrode is nickel oxide (NiOOH) and the negative is a metal hydride such as $La_{1/6}Ni_{5/6}$.

To account for the porosity of electrodes and tortuosity of porous electrodes, the properties of the battery, e.g., diffusion or conductivities, have to be corrected. Tortuosity can be defined as a dimensionless quantity that describes the influence of the morphology of a porous electrode on its effective transport properties (Ebner and Wood 2015).

This correction is known as Bruggeman relation, for example:

$$D_e^{eff} = D_e \varepsilon^{1.5} \qquad 1.2.1$$

Where is the porosity. All effective parameters unless otherwise mentioned follow the Bruggeman relation.

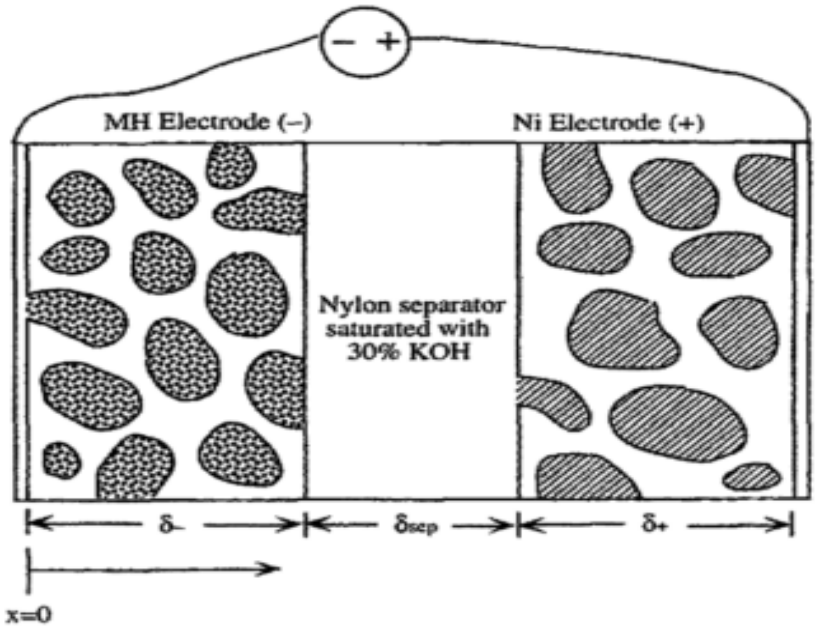

Figure 4. Ni-MH cell, from(B. Paxton and Newman 1997).

1.2.1. Reaction Kinetics

The stoichiometry of the Ni-MH battery is complex; therefore, it has been assumed as a first order reaction, simplified as follows:

$$M + H_2O + e^- \rightleftharpoons MH + OH^- \qquad 1.2.2$$

In (1.2.2) M can represent either a metal hydride or $NiOOH$. For the nickel electrode, the discharge reaction is produced to the right and for the metal hydride to the left, so oxidation occurs in the metal hydride electrode and reduction in the nickel electrode. In that reaction, both electrodes intercalate, i.e. reversible insertion, hydrogen during cycling. This is illustrated in figure 5, in Ni-MH batteries hydrogen is intercalated instead of lithium.

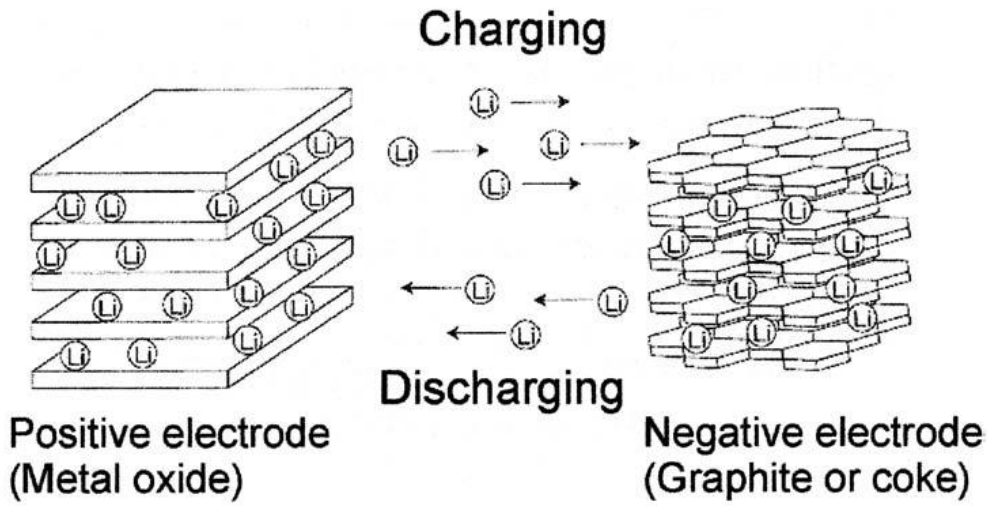

Figure 5. Lithium ions intercalation, from (Kiehne 2003).

Assuming that no side reaction occurs, the net conversion rate for each electrode is:

$$v_{net} = k_f C_f^A - k_b C_b^B \tag{1.2.3}$$

Where k_f and k_b are the rate constants for the forward and backward process respectively, C_f and C_b are the concentrations and the exponents A and B tells the kinetic order of the reactions. Henceforth, a first order kinetics is assumed, so in the further treatment those exponents shall be dropped.

These reactions are heterogeneous, i.e., they only occur at the electrode-electrolyte interface. Thus, the rate can be expressed as:

$$v = \frac{j}{nF} \tag{1.2.4}$$

Where j is the current density (Acm^{-2}).

Therefore, the concentrations shall be expressed as surface concentrations. Now, the net reaction rate in terms of current density is:

$$j = j_f - j_b = nF\left[k_f C_f^A - k_b C_b^B\right] \tag{1.2.5}$$

The rate constant is normally known through experimental data, however, developing the theory of electrode kinetics(Bard and Faulkner 2001) , it can be written via the standard rate constant k^o:

$$k_f = k^o \exp\left[-\frac{\alpha_f F}{RT}\eta\right] \tag{1.2.6}$$

$$k_b = k^o \exp\left[\frac{\alpha_b F}{RT}\eta\right] \tag{1.2.7}$$

η, stands for the overpotential, i.e., a potential that must be overcome to allow charge transfer, it is defined as:

$$\eta = \phi_s - \phi_e - U \qquad 1.2.8$$

And k^o is the standard rate constant. The larger the value of k^o the faster equilibrium will be achieved. α is the transfer coefficient, ranging from zero to unity. It is an experimental measure of the symmetry of the energy barrier, as illustrated in figure 5. The dashed lines in the figure show the shift in the oxidation curve as the potential increases. This means that the transfer coefficient is potential dependent, though, generally it is taken as a constant value, plus, a value of 0,5 is taken when data lacks (Bard and Faulkner 2001).

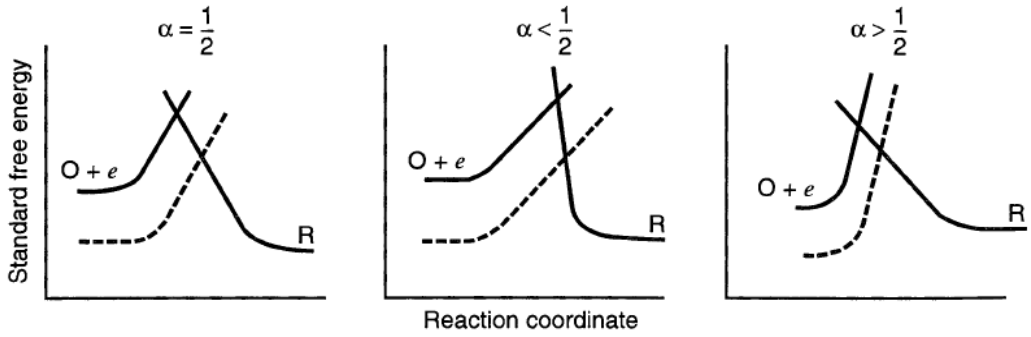

Figure 6. Transfer coefficient as an indicator of the symmetry of the barrier to reaction, extracted from (Bard and Faulkner 2001).

Further information about the theory of electrode kinetics can be found elsewhere (Bard and Faulkner 2001; Rahn and Wang 2012; J Newman 2015).

Combining equations (1.2.5), (1.2.6) and (1.2.7) yields to:

$$j = Fk^o\left[C_f exp\exp\left[-\frac{\alpha_f F}{RT}\eta\right] - C_b exp\left[\frac{\alpha_b F}{RT}\eta\right]\right] \qquad 1.2.9$$

And for the whole battery:

$$j = i^o_{ref} a\left[\mathrm{C_R}\exp\left(\frac{\alpha_a F}{RT}\eta\right) - \mathrm{C_O}\exp\left(-\frac{\alpha_c F}{RT}\eta\right)\right] \qquad 1.2.10$$

Where the subscripts R and O indicates the species involved in reduction and oxidation. That far is imperative to understand that in case to appear more than one reaction in a cell, e.g., in case of side reactions, equation (1.2.10) would represent only one reaction.

Equation (1.2.10) is very important. This, or any variation derived from it, is used in the treatment of almost every problem related to electrochemistry. These results, are known as the Butler-Volmer (BV) formulation of electrode kinetics in honour of the pioneers in the area.

There are many different derivations of the BV equation. In metal hydride batteries, which has insertion kinetics, the BV equation for the reaction current density (j_r) due to production or consumption of a specie usually takes the following form under the assumption that no side reactions occur:

$$\nabla \cdot i = j_r = i^o a \left[\exp\left(\frac{\alpha_a F}{RT}\eta\right) - \exp\left(-\frac{\alpha_c F}{RT}\eta\right)\right] \quad 1.2.11$$

$$i^o = i^o_{ref}\left(\frac{C_s}{C_{s,ref}}\right)^{\alpha_c}\left(\frac{C_e}{C_{s,ref}}\right)^{\alpha_c}\left(\frac{C_o}{C_{o,ref}}\right)^{\alpha_a}\left(\frac{(C_{s,max}-C_s)}{(C_{s,max}-C_{s,ref})}\right)^{\alpha_a} \quad 1.2.12$$

Where, $i^o \; and \; i^o_{ref}$ are the exchange current density and reference exchange current density, a is the interfacial area and $C_s \; and \; C_e$ are the concentrations of the solid and liquid phase.

1.2.2. Transport

Mass transfer, i.e., the movement of material from one point in solution to another, plays a vast role in electrochemical systems. In MH batteries, the species in the solid phase, e.g. H, move to the surface and the electrolyte moves throughout the battery. Depending on the nature that origins the movement, mass transfer is due to:

- Migration: movement owing to a gradient of electrical potential.
- Diffusion: movement due to a concentration gradient.
- Convection: movement caused by a density gradient.

Electrochemical systems are designed so that at least one of the contributions to mass transfer can be neglected. In battery modelling, convection is usually neglected.

In Ni-MH batteries the transport of the solution is modelled according to the concentrated solution theory. For 1:1 electrolyte, electroneutrality stipulates:

$$c = c_- = c_+ \quad 1.2.13$$

Where c is the total concentration of electrolyte, c_- is the concentration of anions and c_+ of cations. This means that there is the same amount of OH^- and K^+, the concentration of solvent is found by knowing that the solution density is equal to the density of the solute plus the density of the solvent:

$$c_o = (\rho - cM_{koh})/M_{H_2O} \qquad 1.2.14$$

Recalling that reactions take place at the electrode surface, conservation of the ion mass involves that the accumulations of ions equals the net input of ions plus the production of ions, yielding to:

$$\varepsilon \frac{dc}{dt} = J - \frac{dN}{dx} \qquad 1.2.15$$

Where the flux (dN/dx) is given by Ficks law of diffusion:

$$N = -D\frac{dc}{dx} \qquad 1.2.16$$

And J, is the flux density of ions from the solid phase:

$$J = -\frac{t^- j_r}{F} \qquad 1.2.17$$

Where j_r is the reaction current density. Combining equations (1.2.15), (1.2.16) and (1.2.17), the conservation of mass for the hydroxide ion is:

$$\varepsilon \frac{dc_e}{dt} = \frac{d}{dx}\left(D_e^{eff}\frac{dc_e}{dx}\right) - \frac{t^- j_r}{F} \qquad 1.2.18$$

Where D_e^{eff} is the effective diffusion coefficient, t^- stands for the transport number, i.e., the fraction of the total current carried in the electrolyte by the ionic species, and c_e is the concentration of the species in the electrolyte. For the solid active particles, the transport is governed by the time-dependent diffusion equation in spherical coordinates:

$$\frac{dc_s}{dt} = D_s\left(\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{dc_s}{dr}\right)\right) \qquad 1.2.19$$

Equation 1.2.19 can be approached from different ways, Paxton and Newman (B. Paxton and Newman 1997) solved it by means of the superposition integral as they stated there was no need of keeping track of the concentration inside the particle. DeVidts (DeVidts, Delgado, and White 1995), solved the equation only once under the supposition that there was no variation alongside the electrode. Gu et al.(Gu, W. B. Wang 1998) solved it in cartesian coordinates and applied an interfacial balance based on the diffusion length to account for the shape of the particles.

1.2.3. Conservation of charge

Electric potential in the solid phase follows Ohm's law:

$$\sigma^{eff}\frac{d^2\phi_s}{dx^2} = j_r \qquad 1.2.20$$

The electrode conductivity is high, from 10^{-1} to 10^4 S/cm since electrodes consist of low-resistance metallic compounds. It depends on the composition of the electrode and how it is fabricated (Rahn and Wang 2012). Those values of conductivity lead to uniform electric potential distribution along the electrode (B. Paxton and Newman 1997).

The electrolyte is also conductive so charge flows, usually without uniform distribution as in the solid phase potential. The charged is carried in the electrolyte by ions, whereas in the solid phase it is by electrons. This charge in the electrolyte is the term of migration in equation 1.2.18 which couples the diffusion of concentration.

The conservation of charge in the electrolyte is written as:

$$k^{eff}\frac{d^2\phi_e}{dx^2} + \frac{d}{dx}\left(\frac{k^{eff}RT}{F}\left(t^- + \frac{c_e}{c^o}\right)\frac{d}{dx}\ln\left(f^{\mp}c_e\right)\right) = -j_r \qquad 1.2.21$$

Where k^{eff} is the effective electrical conductivity and $f^{\mp}$ accounts for the mean activity coefficient. The second term accounts for diffusion of the charged particles associated with concentration gradients (Rahn and Wang 2012). Note that this second term has been found in different expressions in literature, e.g, (Gu, W. B. Wang 1998; B. Paxton and Newman 1997).

The current density term in both equations (1.2.20) and (1.2.21) is the same despite the sign, besides, according to Kirchhoff's law it is possible to distinguish them:

$$I_{cell} = i_s + i_e \qquad 1.2.22$$

Where I is the total current applied to the cell, for the reaction current density:

$$0 = \nabla \cdot i_s + \nabla \cdot i_e = j_{rs} + j_{re} \qquad 1.2.23$$

Equation (1.2.23) means that all the current from the solid phase goes into the electrolyte phase and vice versa, therefore, the subscripts can be dropped.

The current collectors connect to the solid electrodes at the ends of the domain, at $x = 0$ and $x = L$. If the contact resistance is excluded, then the cell voltage is the electric potential at the positive electrode minus the electric potential at the negative electrode:

$$V_{cell} = \phi_s(x = L) - \phi_s(x = 0) \qquad 1.2.24$$

For battery systems with insertion kinetics, such as lithium and MH batteries, the OCP (U) is a function of the local state of charge, which is frequently controlled by solid phase species diffusion,(W. B. Gu and Wang 2000a).

1.2.4. Thermodynamics and cell temperature

The diffusion process slow down at low temperatures, side reactions can be dominant at extreme temperatures, thermal runaway can occur when temperature increases rapidly leading to combustion or explosion. Safety, performance and aging depend critically on temperature; thus, proper modelling is required.

Kinetic rate constants, diffusion coefficients and conductivities depend on temperature. An Arrhenius dependence is often employed (Rahn and Wang 2012), given a property ϕ the relationship established is as follows:

$$\phi = \phi_{ref} \exp\left[\frac{E_{act}^{\phi}}{R}\left(\frac{1}{T_{ref}} - \frac{1}{T}\right)\right] \qquad 1.2.25$$

Where ϕ_{ref} is the property value at the reference temperature (25ºC), E_{act}^{ϕ} is the activation energy of the property.

The open circuit potential is a linear function of temperature (Rahn and Wang 2012) as follows:

$$U = U_{ref} + \left(T - T_{ref}\right)\frac{dU}{dT} \qquad 1.2.26$$

The temperature variation is given by Fourier's law, is:

$$\rho C_p \frac{dT}{dt} = \lambda \frac{d^2T}{dx^2} + q \qquad 1.2.27$$

Where C_p is the specific heat at constant pressure, λ is the thermal conductivity and q is the heat generation rate, whose contributions are:

$$q = q_{irr} + q_{rev} + q_{pol} \tag{1.2.28}$$

The first term is the irreversible heat, the second is the reversible heat and the last is the contribution of polarisation(W. B. Gu and Wang 2000a). The irreversible heat is due to Joule heating:

$$q_{irr} = j_r \eta \tag{1.2.29}$$

The reversible heat accounts for the change of entropy that can be reverted:

$$q_{rev} = j_r T \frac{dU}{dT} \tag{1.2.30}$$

Other losses are due to contact resistance between current collectors, which is neglected, and due to polarisation:

$$q_{pol} = \sigma^{eff} \frac{d^2\phi_s}{dx^2} + k^{eff} \frac{d^2\phi_e}{dx^2} + \frac{k^{eff} RT}{F}\left(t^- + \frac{c_e}{c^o}\right)\frac{d}{dx}\ln\left(f^{\mp} c_e\right)\frac{d\phi_e}{dx} \tag{1.2.31}$$

The first term is the contribution of the solid phase potential and other two correspond to the liquid phase potential.

The term dU/dT in eq. (1.2.30) is the temperature coefficient of an electrode which likewise the OCP depend on experimental evaluation (W. B. Gu and Wang 2000a; Rahn and Wang 2012; J Newman 2015; Rao 1997; Bernardi 1985).

For small cells, if the Biot number is small the first term on the right of eq. 1.2.27 can be dropped. The Biot number tells whether there will be significant temperature differences inside the cell, i.e., if it is small enough it can be considered that there is the same temperature throughout the cell.

$$Bi = \frac{hL_{cell}}{\lambda} \ll 1 \tag{1.2.32}$$

If that condition is fulfilled, eq1.2.27 turns into:

$$\rho C_p \frac{dT}{dt} = \frac{\left(h(T - T_\infty)\right)}{L_{cell}} + q \tag{1.2.33}$$

Where the first term in the right stands for the heat removal from the cell to the surroundings, which in fact is the boundary condition used at the both sides of the cell. Eq. 1.2.33 is called the Lumped Model first developed by Pals and Newman. If this lumped model is taken into account, then, the generation rate is as follows:

$$q = \left(\frac{I_{cell}}{L_{cell}}\right) U - V_{cell} - \frac{T dU}{dT} \qquad 1.2.34$$

In that approach, only reversible and irreversible heat are taken into account. This Lumped model, nonetheless, is not taken into account in this project. Further insight about thermal management of electrochemical batteries can be found in the works carried out by Newman and his co-workers (Rao 1997; Pals and Newman 1995; Bernardi 1985).

1.2.5. Side reactions and aging

Some side reactions remain unnoticed as they do not have long-term effects, others cause permanent degradation contributing to cell's aging (Rahn and Wang 2012). Side reactions need specific conditions in order to trigger; a fine thermal management can avoid or reduce the effects of side reactions.

In Ni-MH batteries the side reactions that can occur are oxygen formation, water hydrolysis, hydrogen absorption into the active material and water corrosion. In order these are:

$$\frac{1}{2} O_2 + H_2O + 2e^- \leftrightharpoons 2OH^- \qquad 1.2.35$$

$$2H_2O + 2e^- \rightarrow H_2 + 2OH^- \qquad 1.2.36$$

The first reaction proceeds to the right in the positive electrode during discharge and to the left in the negative during discharge. While it seems that oxygen evolution is only important during charge and overcharge, it does have effects on the cell performance during discharge(Gu, W. B. Wang 1998).

The second reaction, which is normally not reversible, is the hydrolysis of water which consumes water to form hydrogen, increasing the internal resistance and the electrolyte concentration. Also, hydrogen can be absorbed by the active material, diminishing the performance, and when not absorbed the pressure inside the cell increases. Another effect of hydrogen insertion in the active material is an expansion and contraction of the lattice inducing stress.

Another contribution to aging is the decay of the battery components: electrodes and separator. In time the conductivity of the electrodes diminishes, so does porosity. A change in porosity is associated with active mass loss, i.e., some of the active material no longer participates in the reactions. Capacity fades as well decreasing the capacity of the active particles in the solid phase to intercalate having a direct effect on the porosity variation. There are also changes in stoichiometry associated with the loss the cyclable active material (solid phase species) consumption, e.g., in side reactions.

1.2.6. Ni-MH System characteristics

In order to obtain realistic simulation, it is paramount to foresee its behaviour. From stoichiometry and from figure 4, for instance, it should be expected an increase of solid-phase concentration in the positive electrode and a decrease in the negative during discharge and vice versa during charge. Knowing that, the cell potential is expected to decrease with time during discharge and increase during charge.

At the end of discharge, the nickel electrode shall be fully charged, i.e., $c_s^{Ni} = c_{s,max}^{Ni}$ and the metal hydride fully discharge. That is not true owing to the fact that the capacity of the nickel electrode is less than the MH electrode, this means that the battery is positive electrode limited. For charge process the nickel electrode will be depleted before the MH is entirely charged, this design is to prevent hydrogen evolution in the MH electrode when the cell if fully charged or overcharged.

Likewise, when simulating discharge or charge the initial concentrations in the solid phase should not be taken as $c_s = c_{s,max}\ or\ 0$ but instead $c_s = 0{,}95c_{s,max}$ to prevent this hydrogen evolution (B. Paxton and Newman 1997). Figure 6 presents a simulated discharge curve. There is an initial quick drop followed by a large portion of a shallower potential drop, which is dependent on the discharge rate, and it ends with a drastic drop due to the depletion of the active material in the nickel electrode.

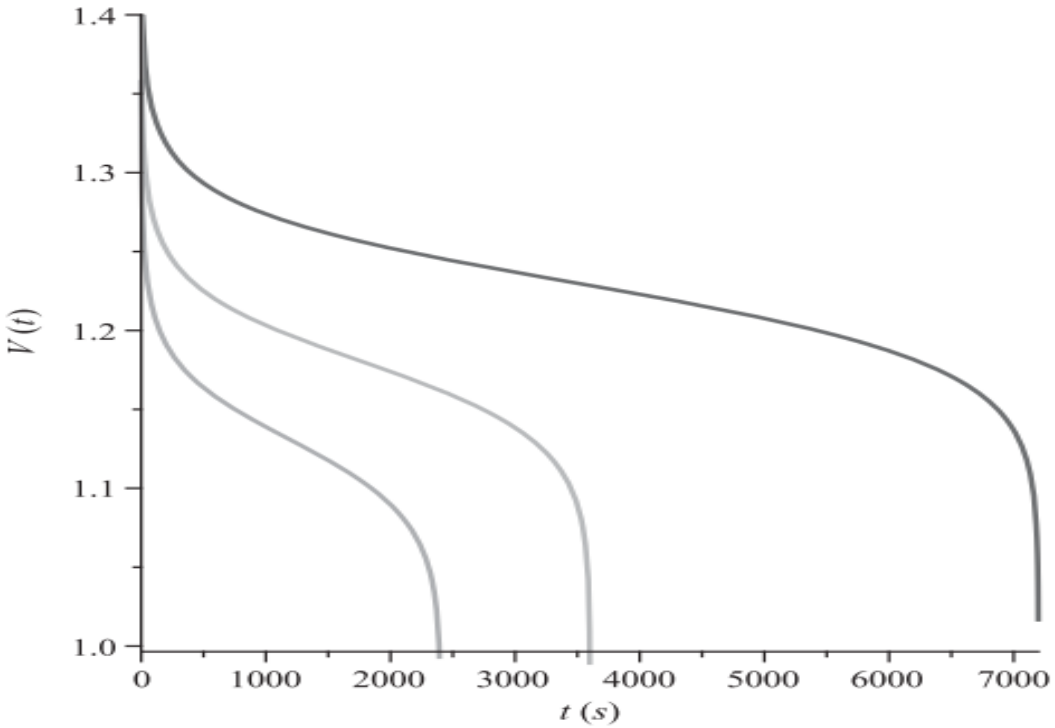

Figure 7. Ni-MH cell discharge voltage, extracted from Battery Systems Engineering

Figure 7, on the other hand presents a simulated charge curve. There is a quick rise at the beginning, followed by a stage of gradual increase of potential and finally a sharp increase as the nickel electrode reaches full charge.

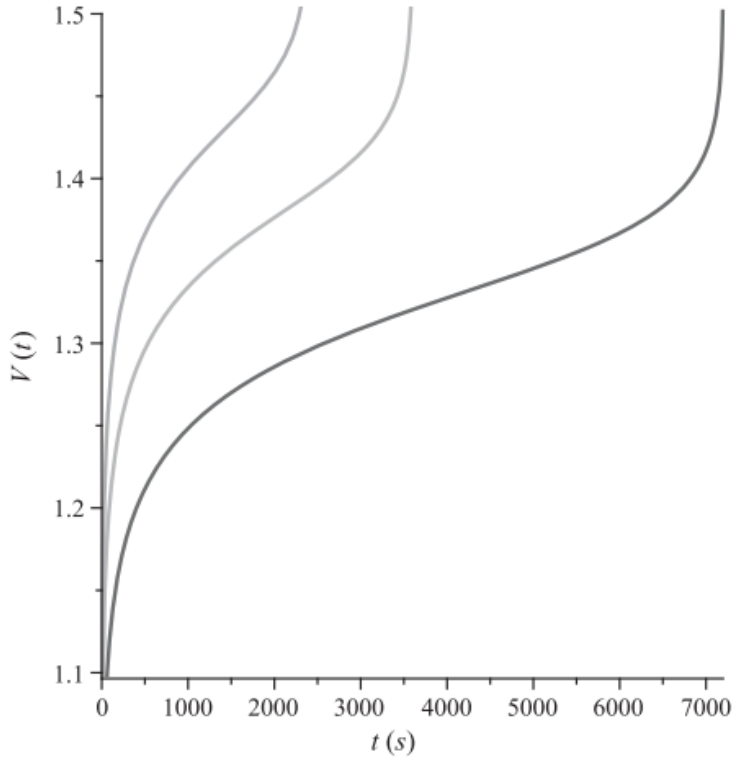

Figure 8. Ni-MH cell charge, extracted from Battery Systems Engineering.

Overall, both curves are similar in shape. Simulations are, therefore, terminated due to the fully depletion/charge of one electrode's active material, nonetheless, before that can occur the battery generally reaches a certain cut off voltage. A cut off voltage is a limit voltage at which the battery is deactivated if trespassed to avoid battery's malfunctioning or damage, it varies depending on the type of battery, manufacturer, etc.

The discharge rate significantly affects the cell behaviour, essentially, the larger the more quickly the potential drop is; not only this but also the initial and final cell voltage is effected by the rate, higher rates mean lower initial voltage and faster reach of the cut off voltage. The discharge rate (C) is the current at which the cell is discharged, a $1C$ discharge rate equals to the current at which the cell theoretically takes one hour to discharge and vice versa for charge.

The theoretical time in hours that a cell takes to discharge is:

$$t = Q/I_{cell} \qquad 1.2.39$$

Where I_{cell} is the applied current density of charge/discharge and Q is the maximum capacity per unit of projected electrode area of an electrode.

In figures 6 and 7 the cell voltage is plotted against time; it is more common though to find it plotted against the depth of discharge (DOD) in discharge or state of charge(SOC) in charge. These quantities define how much the battery is discharged or charged and therefore, one is the inverse of the other, for one unit of increase the DOD is defined as:

$$DOD = \frac{I_{cell}t}{Q} \qquad 1.2.40$$

Chapter 2 Discretization procedures

Diffusion and migration are the main mechanism of mass transfer studied in this work. Following Patankar (Patankar 1980), a numerical 1D approximation has been set up based on finite volumes. The general governing equation is cast into the following form:

$$p\frac{d\phi}{dt} = \frac{d}{dx}\left(\Gamma\frac{d\phi}{dx}\right) + S \qquad 2.1.1$$

For cartesian coordinates and, for spherical coordinates:

$$p\frac{d\phi}{dt} = \frac{1}{R^2}\frac{d}{dr}\left(\Gamma \mathrm{R}^2\frac{d\phi}{dr}\right) + S \qquad 2.1.2$$

Where p is a property of the material (e.g. porosity of the electrode), Γ is a property pertinent to ϕ (e.g. diffusivity coefficient), ϕ represents a general conservable quantity (e.g. concentrations), S is the source term, t is time, R is the radius of a particle and x and r the direction.

2.1. Meshing

A mesh consists in dividing a cell slab with certain properties p, Γ into a number of control volumes.

The algorithm to create such mesh has been provided by Miró (Miró Jané 2014). His algorithm creates a node-centred 1D mesh, i.e. the nodes are located in the centres of the control volumes.

The total number of nodes in the mesh is the sum of all the nodes of the slabs plus two, one for each boundary node. For a single slab, the nodes are spaced equally:

$$dx = \frac{x_f - x_o}{n-1} \qquad 2.1.3$$

Where, n is the number of nodes. The boundary nodes' spacing is half the computed spacing:

$$dx_1 = \frac{dx_1}{2} \qquad 2.1.4$$

$$dx_{end} = \frac{dx_{end}}{2} \qquad 2.1.5$$

The remaining spacing is exactly dx and has to be equally added to the spacing of the interior nodes. Using the subscript i for an arbitrary node, the theoretical position of the nodes is:

$$x_i = x_{i-1} + dx_{i-1} \tag{2.1.6}$$

Then, the position of the boundaries of the control volumes is:

$$x_{vc_i} = x_i + \frac{dx_i}{2} \tag{2.1.7}$$

If multiple slabs are defined, the positions of the nodes need to be recomputed using the control volume positions:

$$x_i = \frac{x_{vc_i} - x_{vc_{i-1}}}{2} \tag{2.1.8}$$

Finally, the resulting mesh computed with Matlab® looks like as follow:

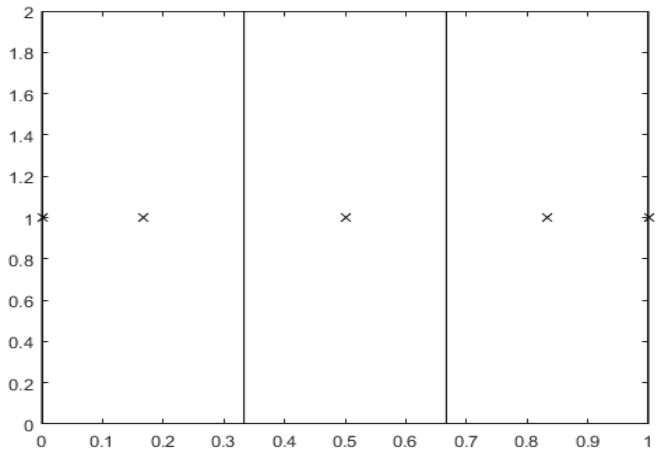

Figure 9. Mesh of longitude unity with 3 nodes.

2.2. Numerical approximation

A finite differences integration technique is used for numerically integrating eqs. 2.1.1 and 2.1.2. According to Patankar (Patankar 1980) it is as follow:

$$\epsilon \int_w^e \int_t^{t+\Delta t} \frac{d\phi}{dt}\ dt\ dx = \int_t^{t+\Delta t} \int_w^e \frac{d}{dx}\left(\Gamma \frac{d\phi}{dx}\right)\ dx\ dt \tag{2.1.9}$$

$$\epsilon \int_w^e \int_t^{t+\Delta t} \frac{d\phi}{dt}\ dt\ dr = \int_t^{t+\Delta t} \int_w^e \frac{1}{R^2} \frac{d}{dr}\left(\mathrm{R}^2 \Gamma \frac{d\phi}{dr}\right)\ dr\ dt \qquad 2.1.10$$

Which respectively yields to:

$$\epsilon \frac{\Delta x}{\Delta t}\left(\phi_p - \phi_p^0\right) = \Gamma\left[\frac{\phi_e - \phi_p}{\Delta x_e} - \frac{\phi_p - \phi_w}{\Delta x_w}\right] \qquad 2.1.11$$

$$\epsilon \frac{\Delta r}{\Delta t}\left(\phi_p - \phi_p^0\right) = \Gamma\left[\frac{R_{p+0,5}^2(\phi_e - \phi_p)}{R_p^2 \Delta r_e} - \frac{R_{p+0,5}^2(\phi_p - \phi_w)}{R_p^2 \Delta r_w}\right] \qquad 2.1.12$$

By putting the above equations in integration coefficients:

$$a_p \phi_p = a_e \phi_e + a_w \phi_w + b \qquad 2.1.13$$

Eq. 2.1.13 stands for the fully implicit scheme. Other schemes, as well as the detailed discretisation can be found in Patankar's. For the above equation, a_p, a_e, a_w are the integration coefficients. ϕ_p is the variable at the current node, ϕ_e and ϕ_w are the variable of the nodes east and west of the current node. The $\Delta x\ and\ \Delta r$ stand for the spatial integration step, respectively, Δt refer to the time integration step.

The source term b, is linearized as it may be function of time, position and the variable. Thus, the following linear approximation is made:

$$b = S_c(t,x) + S_p(t,x)\phi_p \qquad 2.1.14$$

The values of the coefficients of Eq. 2.1.13 are:

Table 1 Values of coefficients of Eq. 2.1.13

Eq. 2.1.15 to 2.1.19	**Eq. 2.1.20 to 2.1.24**
$\boldsymbol{a_e = \frac{\Gamma}{\Delta x_e}}$	$a_e = \frac{\Gamma \mathrm{R}_{p+0,5}^2}{R_p^2 \Delta r_e}$
$\boldsymbol{a_w = \frac{\Gamma}{\Delta x_w}}$	$a_w = \frac{\Gamma \mathrm{R}_{p+0,5}^2}{\mathrm{R}_p^2 \Delta r_w}$
$\boldsymbol{a_p^0 = \frac{\epsilon \Delta x}{\Delta t}}$	$a_p^0 = \frac{\epsilon \Delta r}{\Delta t}$

$\boldsymbol{b = S_c \Delta x + a_p^0 \phi_p^0}$	$b = S_c \Delta r + a_p^0 \phi_p^0$
$\boldsymbol{a_p = a_e + a_w + a_p^0 - S_p \Delta x}$	$a_p = a_e + a_w + a_p^0 - S_p \Delta r$

To reduce this equation to the steady-state discretization equation it is required to set $\Delta t \to \infty$.

2.2.1. Source term linearization

As explained, the source term can depend on ϕ. Therefore, the dependence must be expressed in a linear form, already given by Eq. 2.2.14. This is done because the framework allows only linear dependence and a linear dependence is better than treating it as a constant.

When S is a nonlinear function of ϕ, linearization must be made i.e. the values of S_c and S_p may themselves depend on ϕ. Therefore, recalculation after each iteration with new values of ϕ is expected. Likewise, such linearization should be a good representation of the $S \sim \phi$ relationship.

Recalling Eq. 2.1.14:

$$S = S_c + S_p \phi \qquad 2.1.25$$

The following relationship is stablished:

$$S_c = \left[S(\phi) - \frac{dS}{d\phi} \phi \right] \qquad 2.1.26$$

$$S_p = \frac{dS}{d\phi} \qquad 2.1.27$$

Other ways of linearizing the source terms are possible and can be found in Patankar (Patankar 1980). Notwithstanding an example from Patankar is set:

Example. Given $S = 5 - 4\phi$. Some linearizations are:

1. $S_c = 4,\ S_p = -4$. This is the most obvious and the recommended.
2. $S_c = 5 - 4\phi^*, S_p = 0$. This approach throws the whole S into S_c. This practice is not impracticable and perhaps the only choice should the expression of S be highly complicated.
3. $S_c = 5 + 7\phi^*, S_p = -11$. This is a steeper $S \sim \phi$ relationship proposal than the one given. The convergence of the iterations will slow down, however, if there are other nonlinearities in the problem the slowdown may be welcome.

2.2.2. The four basic rules

Pantankar presents four basic rules to ensure stability and proper behaviour of the described method:

Rule 1: Consistency at control-volume faces. The flux that leaves one control volume must be identical to the flux that enters the next control volume.

Rule 2: Positive coefficients. To ensure a physically realistic solution.

Rule 3: Negative-slope linearization of the source term. To ensure stability and that physically unrealistic solutions do not arise, the coefficient S_p must be negative.

Rule 4: Sum of the neighbour coefficients. The coefficient a_p must be the sum of the other coefficients. The method used fulfils automatically this rule.

2.3. Boundary conditions

Two types of boundary conditions are considered:

1. Given boundary variable.
2. Given boundary flux.

The first condition is written as:

$$\phi(t, x = 0, L) = \phi^* \qquad 2.1.28$$

To arrange the solver to give ϕ^* as a solution for the boundary is done by modifying the coefficients as follow:

$$a_i = 1 \qquad 2.1.29$$

$$a_p = 0 \qquad 2.1.30$$

$$b = \phi^* \qquad 2.1.31$$

a_i is the adjacent node to the boundary, being W or E. The other boundary node as it does not exist, is set to zero.

The second condition is written as:

$$-\Gamma \frac{d\phi}{dx} = J \qquad 2.1.32$$

Eq. 2.1.32 is numerically integrated in the half control volume near the boundary as shown:

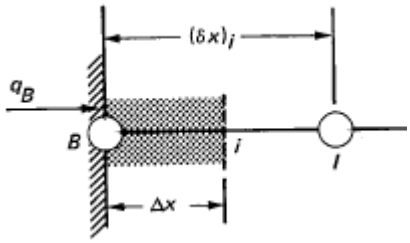

Figure 10 Half control volume near the boundary (from (Patankar 1980)).

The effect of this derivation results in the addition of a parameter in the source term coefficient, again the full derivation can be found in Patankar's. Henceforth, the coefficients for this case are:

$$a_i = \frac{\Gamma}{\Delta x_i} \qquad 2.1.33$$

$$a_p = a_i + ap^0 - S_p \Delta x \qquad 2.1.34$$

$$b = S_c \Delta x + J + a_p^0 \phi^0 \qquad 2.1.35$$

2.4. Solver

The solver used in this work is the Thomas algorithm, also called TDMA (Tri Diagonal Matrix Algorithm). It uses Gauss-elimination procedure that is delightfully convenient for diagonal matrices, the discretisation explained in this chapter leads to all nonzero coefficients aligned in three diagonals inside the coefficient matrix, furthermore, the required computer time is only proportional to the number of nodes (N) rather than N^2 or N^3 (Patankar 1980).

The computed matlab algorithm for this solver is also provided by Miró (Miró Jané 2014). In this case, only a summary of the algorithm is shown in order to provide some information as the full derivation can be found in Pantankar's.

Compute P_1 and Q_1 from:

$$P_1 = \frac{a_{E_1}}{a_{P_1}} \quad 2.1.36$$

$$Q_1 = \frac{b_1 + a_{W_1}\phi_0}{a_{P_1}} \quad 2.1.37$$

Using the recurrence:

$$P_i = \frac{a_{E_i}}{a_{P_i} - a_{W_i}P_{i-1}} \quad 2.1.38$$

$$Q_i = \frac{b_i + a_{W_i}Q_{i-1}}{a_{P_i} - a_{W_i}P_{i-1}} \quad 2.1.39$$

Obtaining P_i and Q_i for $i = 2,3, \dots, N$.

Setting $\phi_N = Q_N$.

Use

$$\phi_i = P_i\phi_{i+1} + Q_i$$

To compute the variable for $i = N-1, N-2, \dots, 3,2,1$.

2.4.1. Solution procedure

The following steps are taken to solve this sort of problem for a single time step:

1. Suppose ϕ^*.
2. Make $\phi = \phi^*$.
3. Compute any property, such diffusion coefficients, using ϕ.
4. Compute the coefficients a_e, a_w, a_p, b.
5. Solve the system of equation.
6. If $\phi - \phi^* < \varepsilon$, where ε is an arbitrary small value, then convergence is reached and proceeds to the following time step. If not, return to step 2.

This approach is needed when the problem is not linear, i.e. any properties depend on ϕ or the boundary conditions force any nonlinearity.

In the overall iterative scheme employed for handling nonlinearities, it is sometimes desirable to speed up or to slow down the variable changes throughout the iterations, these processes are called over-relaxation and under-relaxation respectively.

By introducing a relaxation factor k between 0 and 1, it is possible to slow down the process and avoid divergence problems in the iterative solution of nonlinear equations, i.e values of ϕ stay closer to ϕ^*. In the scheme described previously, it is possible to introduce the factor after step 6, then the new step 2 would be:

2. Make $\phi = k\phi^*$.

It should be noted that when using relaxation schemes, they must fulfil the original discretized equation. There are no general rules for choosing the best value of k, and a suitable value must be found by trial and error (Patankar 1980).

2.5. Validation

It is important to differentiate between verification of calculations, i.e. errors in stations and code verification, i.e. errors evaluations from known solutions. The code is validated through the Method of the Manufactured Solutions (MMS). The MMS was first proposed by Roache (Roache 2002)and later by Salari and Knupp (Salari, Kambiz, Knupp 2000).

The MMS consists of taking an example solution $U(x,t)$ before specifying the governing equations, that solution must satisfy the problem's boundary conditions for which has been defined. Then, this solution is introduced as the source term and the case is solved numerically, the numerical solution obtained should be equal to the selected example.

To validate a general case, two meshes shall be analysed. One mesh must set $dt \to \infty$ to eliminate the transient term. For that case, which corresponds to a steady state, the picked solution is:

$$U(t,x) = sin1sinx - \sin^2 x \qquad 2.1.40$$

With the boundary conditions:

$$U(0,x) = 0 \qquad 2.1.41$$

$$U(t,0) = 0 \qquad 2.1.42$$

$$U(t,1) = 0 \qquad 2.1.43$$

Recalling the general governing equations with the transient term set to infinity:

$$S = -\frac{d}{dx}\left(\Gamma\frac{d\phi}{dx}\right) \qquad 2.1.44$$

$$S = -R^2\frac{d}{dr}\left(\Gamma\frac{d\phi}{dr}\right) \qquad 2.1.45$$

The source term in eq. 2.1.44 is the second derivative, therefore:

$$S = -\Gamma\frac{d^2U}{dx^2} = \Gamma sin1 sinx + 2\Gamma cos2x \qquad 2.1.46$$

Similarly, for eq.2.1.45:

$$S = -\Gamma R^2\frac{d^2U}{dr^2} = R^2\Gamma sin1 sinr + 2\Gamma R^2 sin1 cos2r \qquad 2.1.47$$

To analyse the temporal mesh, a known working value of dx is chosen. For this case, the selected solution is:

$$U(t,x) = x(1-x)t \qquad 2.1.48$$

With the same boundary conditions, eqs. 2.1.41 to 2.1.43.

Therefore, the source terms to be input into the numerical solver are:

$$S = p\frac{dU}{dt} - \Gamma\frac{d^2U}{dx^2} = \mathrm{px}(1-\mathrm{x}) + 2\Gamma\mathrm{t} \qquad 2.1.49$$

$$S = pR^2\frac{dU}{dt} - R^2\frac{d}{dr}\left(\Gamma\frac{d\phi}{dr}\right) = pR^2x(1-x) + 2\Gamma t \qquad 2.1.50$$

The selected solutions are the same as picked in previous works, it is the belief of the author that given the solutions have already been useful to other authors (Miró Jané 2014), there is no need to search for others. Also, for the spherical coordinates case x turns into r.

The error E between the numerical and exact solution is as follows:

$$E = max(f(\Delta)_{num} - f_{exact}) \qquad 2.1.51$$

For a well-behaved problem, such as the finite differences method, the following relationship holds (Roache 2002; Salari, Kambiz, Knupp 2000) :

$$E = C\Delta^p + \varepsilon(\Delta^2) \qquad 2.1.52$$

Where C is a constant and p is the theoretical order of the method. By refining the mesh, i.e. increasing the number of nodes, the logarithm of the error versus the logarithm of the mesh refinement must hold linear relationship of order p. By selecting two different mesh refinements and taking logarithms:

$$\log\left(\max(f_1(\Delta_1)_{num} - f_{1_{exact}})/\max(f_2(\Delta_2)_{num} - f_{2_{exact}})\right) = plog\left(\frac{\Delta_1}{\Delta_2}\right) \qquad 2.1.53$$

This can be done with time differentials as well, which yields to time refinement.

2.5.1. Results from validation

The MMS is run by both mesh in spacing dx, dr and time dt. The error is plot versus the mesh refining to assess the quality of the mesh and linear regression is made to obtain the order of approximation.

For the first case, both cartesian and spherical, a slope of 2 is expected as a result of an approximation of a second order derivative.

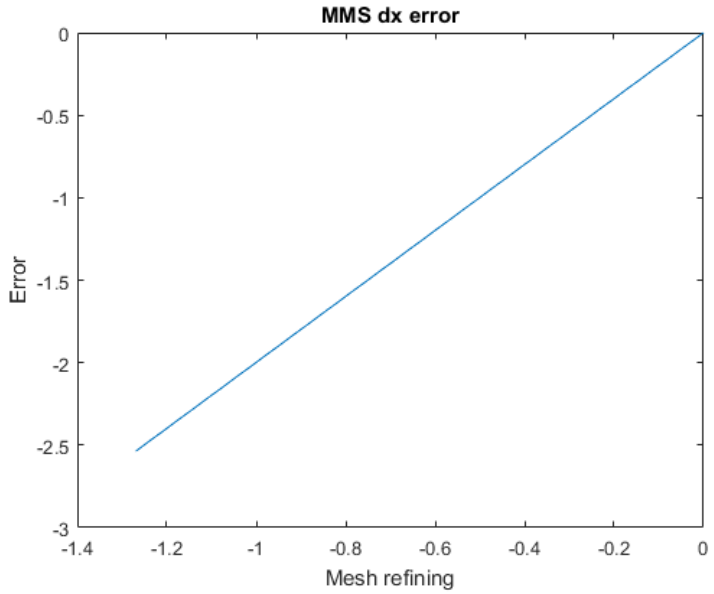

Figure 11.MMS error plot for the cartesian spacing mesh, p=1,999.

The slope of the curve is 1,999, the order of approximation is as expected and really close to previous works (2,03 in Miro's work) thus, it is verified that the code runs smoothly for steady state case.

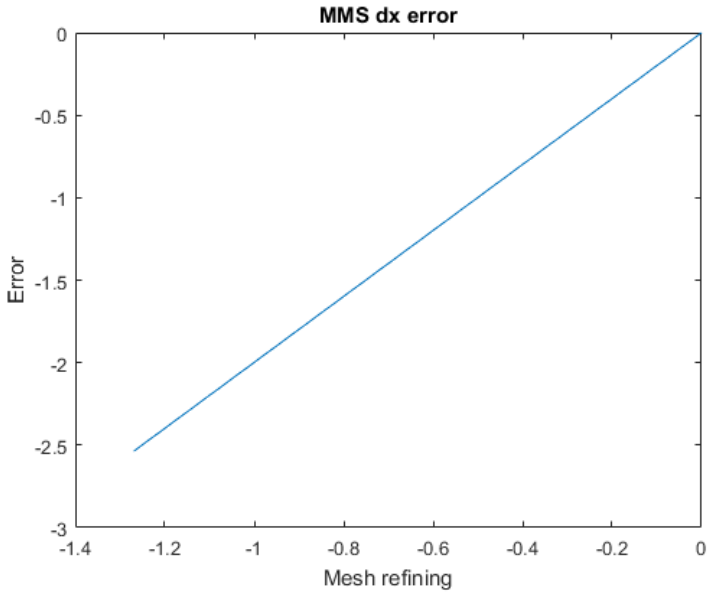

Figure 12. MMS error plot for the spherical spacing mesh, p=1,999.

For the spherical coordinates, the slope of the curve is again 1,999 as expected, the code is verified as well for this case. Resemblances between plots is due to the same mesh refining.

Regarding the time step refinement, a slope of 1 is expected as a result of a first order approximation of a first derivative.

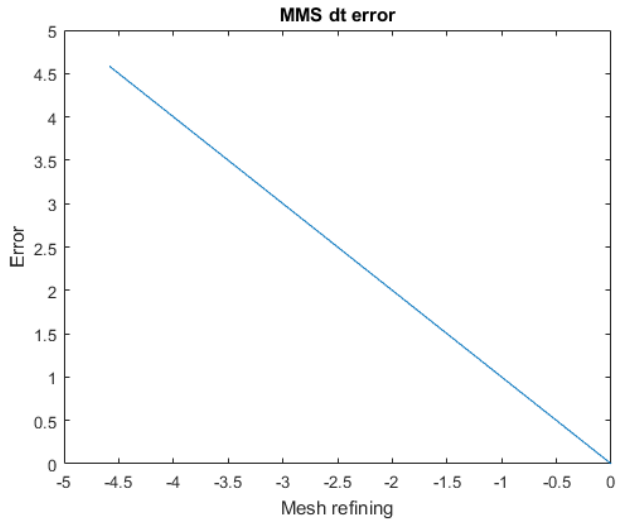

Figure 13.MMS error plot for the cartasian time mesh, p=1.

The slope of the curve is 1, the order of approximation is as expected and really close to previous works (1,24 in Miro's work) thus, it is verified that the code runs smoothly for the transient case.

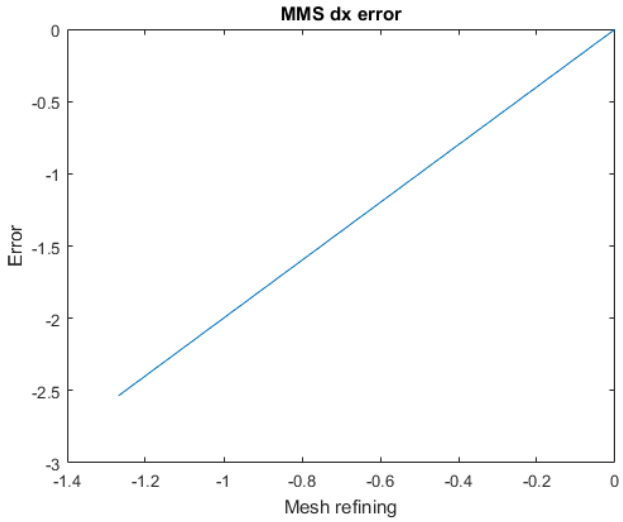

Figure 14. MMS error plot for the spherical time mesh, p=0,9986.

For the spherical coordinates the slope of the curve is 0,9986 again as expected, therefore, the code is verified for this case.

Chapter 3 1D Ni-MH Model

In this chapter the nickel metal hydride battery 1D model is formulated, i.e., assumptions, governing equations, parameters and solving procedure. It is important to note that the governing equations are related to those presented in chapter 1. To give further insight on the solving procedure, the MATLAB toolbox is presented. Finally, the model is validated against other models presented in literature and further assessment is carried out.

The battery consists of three different slabs: a metal hydride electrode of length L_{MH}, a separator of length L_{sep} and a nickel electrode of length L_{Ni}. Those slabs are computed in a single mesh of L_{cell} and each slab is divided into the same number of nodes for discretisation. A separated mesh is created for the active particles in the electrodes. Recalling figure 4, the system consists on a solid phase and a liquid phase owing to the fact that the electrodes are porous and flooded with electrolyte except for the separator that there is no solid. Finally, the active particles in the porous electrodes are taken as spheres.

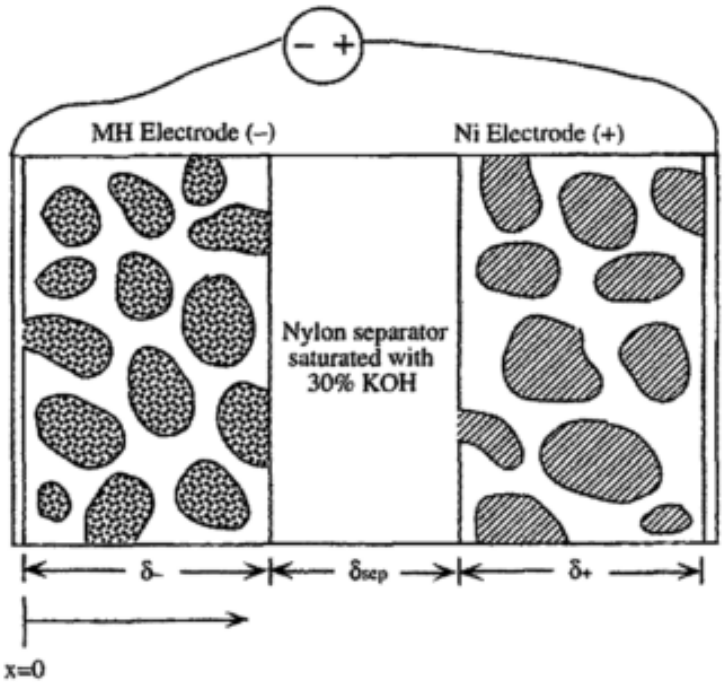

Figure 15. Ni-MH cell replica of figure 4, from(B. Paxton and Newman 1997).

3.1. Assumptions

The assumptions upon this model is based are made for the purpose of rigorously comparing with previous works:

1. Each electrode is a two-phase system consisting of a solid matrix and electrolyte as shown in figure 15.
2. Porosity variation is neglected.
3. Side reactions (e.g. oxygen evolution) are neglected (B. Paxton and Newman 1997).
4. Convection contribution to transport is neglected, leaving it only to migration and diffusion (B. Paxton and Newman 1997; Gu, W. B. Wang 1998; DeVidts, Delgado, and White 1995).
5. Particles in the solid phase are assumed to be spherical with radius R (B. Paxton and Newman 1997).
6. There are no gradients in the angular directions inside and on the surface of the spherical particles (DeVidts, Delgado, and White 1995).
7. There is no pressure variation alongside the cell (B. Paxton and Newman 1997; DeVidts, Delgado, and White 1995).
8. Contact resistance is neglected (DeVidts, Delgado, and White 1995; Gu, W. B. Wang 1998; B. Paxton and Newman 1997).

3.2. Governing equations

3.2.1. Species governing equations

There are two species participating in charge-transfer reaction: an ionic species present in the electrolyte OH^- and proton or atomic hydrogen, H, present in the solid phase of the electrodes (Nickel and MH).

Under the assumptions in section 3.1 the convective term vanishes. For the liquid electrolyte, the species concentration OH^- is solved denoting the subscript e for this species, according to section 1.2.2, this yields:

$$\varepsilon \frac{dc_e}{dt} = \frac{d}{dx}\left(D_e^{eff} \frac{dc_e}{dx}\right) - \frac{t^- j_r}{F} \qquad 3.2.1$$

And the solvent concentration is obtained as follows:

$$c_o = \frac{\rho - cM_{koh}}{M_{H_2O}} \quad 3.2.2$$

Where the diffusion follows the Bruggeman relation.

At the outer boundaries of the cell, a zero flux condition is applied:

$$\frac{dc_e}{dx} = 0 \; at \; x = 0, L \quad 3.2.3$$

The specie in the solid phase is denoted by the subscript s, the concentration is governed by the simple time dependent diffusion equation in spherical coordinates:

$$\frac{dc_s}{dt} = D_s \left(\frac{1}{r^2} \frac{d}{dr} \left(r^2 \frac{dc_s}{dr} \right) \right) \quad 3.2.4$$

At the centre of the particles, there is no flux:

$$\frac{dc_s}{dr} = 0 \; at \; r = 0 \quad 3.2.5$$

Recalling that the reaction takes place at the interface between the liquid and solid phase, at the surface of the particle the flux is equal to the reaction rate:

$$-D_s \frac{dc_s}{dr} = \frac{j_r}{a_{se} F} \quad 3.2.6$$

Where a_{se} is the interfacial area of the nickel electrode or the metal hydride.

To account for the variation of solid phase concentration alongside the electrode, the interfacial balance approach proposed by (Gu, W. B. Wang 1998) is used:

$$c_{se} = c_s[r = R] - \frac{r_s \, j}{D_s a_{se} F5} \quad 3.2.7$$

3.2.2. Charge conservation equations

The electrical potential in the electrolyte is given according to section1.2.3:

$$k^{eff}\frac{d^2\phi_e}{dx^2}+\frac{d}{dx}\left(\frac{k^{eff}RT}{F}\left(t^-+\frac{c_e}{c^o}\right)\frac{d}{dx}\ln\left(f^{\mp}c_e\right)\right)=-j_r \qquad 3.2.8$$

Where k^{eff} follows the same Bruggeman relation as diffusion. Zero flux boundary conditions are applied on both sides of the cell:

$$\frac{d\phi_e}{dx}=0\ at\ x=0,L \qquad 3.2.9$$

The potential in the solid phase is given by Ohm's law:

$$\sigma^{eff}\frac{d^2\phi_s}{dx^2}=j_r \qquad 3.2.10$$

σ^{eff} follows the Bruggeman relation. In the negative electrode, a reference potential is set, normally an electric ground:

$$\phi_s=0\ at\ x=0 \qquad 3.2.11$$

At the positive electrode, the current density is specified.

$$-\sigma^{eff}\frac{d\phi_s}{dx}=I_{cell} \qquad 3.2.12$$

All the governing equations are coupled by the density current term that takes the form of the Butler-Volmer's equation, for insertion kinetics as explained in section1.2.1:

$$j_r=i^o a\left[\exp\left(\frac{\alpha_a F}{RT}\eta\right)-\exp\left(-\frac{\alpha_c F}{RT}\eta\right)\right] \qquad 3.2.13$$

With i^o as:

$$i^o=i^o_{ref}\left(\frac{C_s}{C_{s,ref}}\right)^{\alpha_c}\left(\frac{C_e}{C_{s,ref}}\right)^{\alpha_c}\left(\frac{C_o}{C_{o,ref}}\right)^{\alpha_a}\left(\frac{(C_{s,max}-C_s)}{(C_{s,max}-C_{s,ref})}\right)^{\alpha_a} \qquad 3.2.14$$

The overpotential η is:

$$\eta=\phi_s-\phi_e-U \qquad 3.2.15$$

Where U is the open circuit potential that is SOC dependent , this work uses SOC dependent values for U, which can be found in (COMSOL 2011).

3.2.3. Temperature variation

Temperature changes mainly due to heat dissipation. The governing equation for the temperature inside the cell according to section1.2.4 is:

$$\rho C_p \frac{dT}{dt} = \lambda \frac{d^2T}{dx^2} + q \qquad 3.2.16$$

The heat source is due to the contributions of ohmic heats in both the matrix and the electrolyte, irreversible heat and reversible heat. Putting all terms together, that is:

$$q = \sigma^{eff} \frac{d\phi_s}{dx}\frac{d\phi_s}{dx} + k^{eff} \frac{d\phi_e}{dx}\frac{d\phi_e}{dx} + \frac{k^{eff}RT}{F}\left(t^- + \frac{c_e}{c^o}\right)\frac{d}{dx}\ln\left(f^{\mp}c_e\right)\frac{d\phi_e}{dx}\, j\eta + jT\frac{dU}{dT} \qquad 3.2.17$$

The heat transfer with the exterior of the cell occurs by convection with the air. Therefore, the boundary conditions at both sides of the cell are:

$$-\lambda \frac{dT}{dx} = h(T - T_\infty)\ at\ x = 0, L \qquad 3.2.18$$

Given the lack of data regarding the dU/dT parameter, it has been taken as a constant value.

Equations 3.2.8,3.2.10 comprise the isothermal model, the variables to be solved are c_e, c_s, ϕ_s and ϕ_e. This is in accordance with (DeVidts, Delgado, and White 1995; Gu, W. B. Wang 1998; B. Paxton and Newman 1997). It is also possible to manipulate the equations in order to solve η and i instead of ϕ_s and ϕ_e(B. Paxton and Newman 1997; J Newman 2015).

3.2.4. Entropy

One method to assess the battery's decay is via the irreversible entropy, according to literature entropy is indeed an indicator for degradation (Bryant 2014; Cuadras et al. 2015; Cuadras, Ovejas, and Quilez 2013). Likewise, the irreversible heat, the irreversible local entropy at each point of the battery, i.e., at each point of the discretised battery, is:

$$\dot{S}_l = \frac{j_r \eta}{T} \qquad 3.2.19$$

To assess the change of entropy in time, the whole battery has been approached as control volume:

$$\dot{S} = \frac{I_{cell}}{T_{mean}} \frac{1}{L_{cell}} \int_0^{L_{cell}} \eta dx \qquad 3.2.20$$

Where the integral of the overpotential is done to take the mean value, and expecting a small enough Biot number the temperature can be taken as the mean value, otherwise, it should be approached in the same way as the overpotential.

3.3. Parameters

The parameters and correlations for the isothermal case are all summarised in table 2. Regarding the solid phase conductivity, many authors have proposed values without a general consensus, thus, constant estimated values for electric conductivity are used (B. Paxton and Newman 1997). The initial concentration of electrolyte is $6{,}91\ mol/cm^3$, corresponding to 30% weight in water. The initial concentration in the solid particle is set to a 95% SOC of its theoretical maximum and set to 95% in the metal hydride electrode, according to chapter 1. The transference number of OH^- is $0{,}22$. The capacity of the cell is the same as used in Paxton's work, $43{,}4\ mAh/cm^2$.

Table 2. Parameters used in the simulations

Metal hydride electrode		
$\boldsymbol{L_{mh}}$	$350\ \mu m$	
$\boldsymbol{R}$	$1{,}5\ \mu m$	
$\boldsymbol{\varepsilon_e}$	$0{,}396$	(B. Paxton and Newman 1997)
$\boldsymbol{\varepsilon_s}$	$0{,}481$	
$\boldsymbol{a}$	$3\varepsilon_e/R$	

$\boldsymbol{\alpha_a}$	0,25	
$\boldsymbol{\alpha_c}$	0,54	
$\boldsymbol{i^o_{ref}}$	0,785 mA/cm^2	
$\boldsymbol{c_{s,max}}$	0,1025 mol/cm^3	
$\boldsymbol{c_{s,ref}}$	$c_{s,max}/2$	
$\boldsymbol{\sigma}$	1000 S/cm	
$\boldsymbol{D_s}$	$2 \cdot 10^{-8}\ cm^2/s$	
Nickel electrode		
$\boldsymbol{L_{ni}}$	843 μm	
$\boldsymbol{R}$	2,5 μm	
$\boldsymbol{\varepsilon_e}$	0,507	
$\boldsymbol{\varepsilon_s}$	0,481	
$\boldsymbol{a}$	$3\varepsilon_e/R$	
$\boldsymbol{\alpha_a}$	0,13	
$\boldsymbol{\alpha_c}$	0,074	(B. Paxton and Newman 1997)
$\boldsymbol{i^o_{ref}}$	0,104 mA/cm^2	
$\boldsymbol{c_{s,max}}$	0,0383 mol/cm^3	
$\boldsymbol{c_{s,ref}}$	$c_{s,max}/2$	
$\boldsymbol{\sigma}$	28 S/cm	
$\boldsymbol{D_s}$	$3{,}4 \cdot 10^{-8}\left[1-\left(\frac{c_s}{c_{s,max}}\right)\right]^2$	
Separator		

L_{sep}	$250\ \mu m$	(B. Paxton and Newman 1997)
ε_e	0,5	Estimated

Correlations

$\rho\left[{}^{g}/_{cm^3}\right] = 1,001 + 47,52c_e - 776,22c_e^2$	(B. Paxton and Newman 1997)
$D_e\left[{}^{cm^2}/_{s}\right] = 2,8509 \cdot 10^{-5} - 2,9659 \cdot 10^{-4}c_e^{0,5} + 0,013768c_e - 0,14199c_e^{1,5} + 0,42661c_e^2$	
$k[{}^{S}/_{cm}] = 0,02325 + 210,95c_e - 22077c_e^2 + 6,2907 \cdot 10^5 c_e^3$	
$f^{\pm} = 1,004 - 36,23c_e^{0,5} + 1374,3c_e - 17850,7c_e^{1,5} + 55406c_e^2 + 7,16856 \cdot 10^5 c_e^{2,5}$	

Thermal parameters

T_{ref}	298,15 K	Estimated
T_{∞}	298,15 K	
λ_{MH}^{m}	1,16 W/mK	(Wu, Wang, and Wan 1998)
λ_{sep}^{m}	0,22 W/mK	
λ_{Ni}^{m}	1,14 W/mK	
λ_f	0,57 W/mK	
$C_{p,MH}$	0,35 J/gK	(W. B. Gu and Wang 2000a)
$C_{p,Ni}$	0,88 J/gK	
$C_{p,sep}$	1,9 J/gK	
$C_{p,e}$	3,2 J/gK	
ρ_{Ni}	3,55 g/cm^3	

$\boldsymbol{\rho_{sep}}$	$7{,}49\ g/cm^3$	
$\boldsymbol{\rho_{MH}}$	$0{,}9\ g/cm^3$	
$\left(\boldsymbol{dU}/\boldsymbol{dT}\right)_{\boldsymbol{ref}}$	$-1{,}125\ mV/K$	
$\boldsymbol{E_{act,k}}$	$13\ kJ/mol$	
$\boldsymbol{E_{act,MH}}$	$30\ kJ/mol$	
$\boldsymbol{E_{act,Ni}}$	$20\ kJ/mol$	
$\boldsymbol{E_{act,D^{OH}}}$	$14\ kJ/mol$	
$\boldsymbol{E_{act,D^{H}}}$	$9.62\ kJ/mol$	
$\boldsymbol{\lambda = \lambda^m(1-\varepsilon) + \lambda^f \varepsilon}$		(Wu, Wang, and Wan 1998)

3.4. Numerical procedures

The set of equations is solved via the numerical procedure explained in Chapter 2 implemented in Matlab®, each equation is cast in the general form explained there. That is:

$$p\frac{d\phi}{dt} = \frac{d}{dx}\left(\Gamma\frac{d\phi}{dx}\right) + S \qquad 3.2.21$$

In charge-conservation equations the transient term vanishes, this is done by setting a high dt (e.g. 10^{99}). The source term is treated as explained in section 2.2.1. For the electrolyte species concentration, it is indeed possible to set all the source term into S_c, nonetheless, for the charge-conservation and temperature source terms it is compulsory to linearize the source term due to the strong non-linear form of the Butler-Volmer equation.

Gu and his co-workers (W. B. Gu and Wang 2000a; Gu, W. B. Wang 1998) also discretised the equations into this fashion. To help preventing the arousal of non-linarites, a relaxation factor of 0,5 is used.

The cell is divided into three different slabs, one for each electrode plus another for the separator. The typical number of nodes is 60 and time steps range from 10 to 20 seconds (Gu, W. B. Wang 1998). In

this work 100 nodes have been used for each slab as well as for the spherical particles, time steps varying from 5 to 20 seconds have been used.

The set of equations is solved and convergence is reached when the error between two different iterations is less than 10^{-9}.

3.5. Matlab® computational toolbox

The Matlab files that comprised the battery toolbox are summarised in table 3 and the algorithm of the different modules can be found in the Appendix-A.

Table 3. Battery Toolbox Matlab files.

ONEDMHBATTERY	Mail algorithm to simulate the battery.
LIP (Loop internal parameters)	Calculates the parameters that need recalculation at each loop such as the thermal dependences.
Currentdistribution	Computes the Butler-Volmer equation.
PCS (Potential Coupled Solver)	Solve the charge transfer equations.
CCS (Concentration Coupled Solver)	Solve both the electrolyte concentration and solid-phase concentration equations.
HeatTransfer	Solve the heat balance inside the cell.
MESH1D	Creates the mesh and the inputs are put into proper form.

Other files used are MESHNC1D and MESHPLOT1D are lent by (Miró Jané 2014), the first creates a finite difference node centred 1D mesh of a series of different material slabs of longitude L and n nodes inside which is included inside MESH1D, the latter is used to plot the first one. The first one modified to suit the needs of spherical coordinates is named MESHNC1DY. Discretization and resolution of the governing equations done according to chapter 2 are carried out in Diffusion1D which includes the TDMA solver.

To avoid massive use of memory data entry, meshing and the solving subroutine have been separated intro three different matlab files. The process (figure 15) starts with entering the different parameters (initial concentrations, etc.) and the meshing options, i.e., number of slabs, nodes of each slab and longitude of the mesh. By declaring different slabs, the inputs must be converted into 1x1 matrixes. Solver options include time step, solver tolerance, final time, relaxation factor and maximum number of iterations for convergence.

Since the different functions needs an important amount of data, a script file has been created with all the different parameters already with needed for the simulation, it has been tried to avoid arcane notation to ease future user's comprehension.

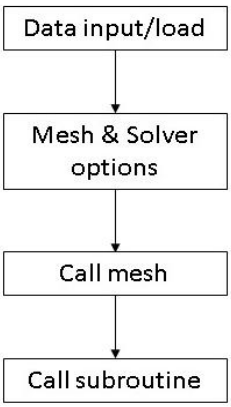

Figure 16. Matlab® algorithm steps.

The solving subroutine is a double *for* loop, one for time and another for convergence in a time step. The *for* loop is desired over while loop to avoid infinite loops. To achieve convergence is paramount to solve the equations by blocks, i.e., cast them in different loops (figure 17). Solving the source term j_r, which appears in all governing equations, properly is most important since it depends on the overpotential and on temperature, thus, it is the main reason of non-linarites.

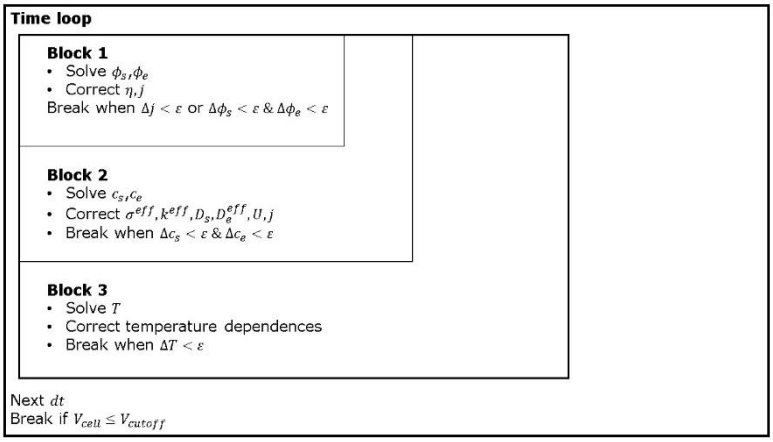

Figure 17. Coupling Scheme.

Each block includes the other, e.g. when block 3 is reached and a new value of T is obtained the algorithm goes back to block 1, this results in increasing the stability, alas computational time is as well increased. Despite that fact, using a standard pc (i7-3630QM @2.4GHz) it takes 94 seconds to solve the problem for a 1C discharge, which is a higher computational time compared with other works(Gu, W. B. Wang 1998). The solving algorithm also includes key stop calls to prevent unrealistic solutions and divergence, i.e. the subroutine stops should the following events occur: cut off voltage is reached, negative concentrations arise, adimensional concentrations get values higher than unity or less than zero.

This coupled algorithm is needed since through the thermal balance new values of the temperature dependent parameters are obtained, and the variables are then needed to be updated. This is in agreement with the electrochemical-thermal coupled modelling approach given by (W. B. Gu and Wang 2000a), figure 18 is a replica from their diagram that shows the relationships between both models.

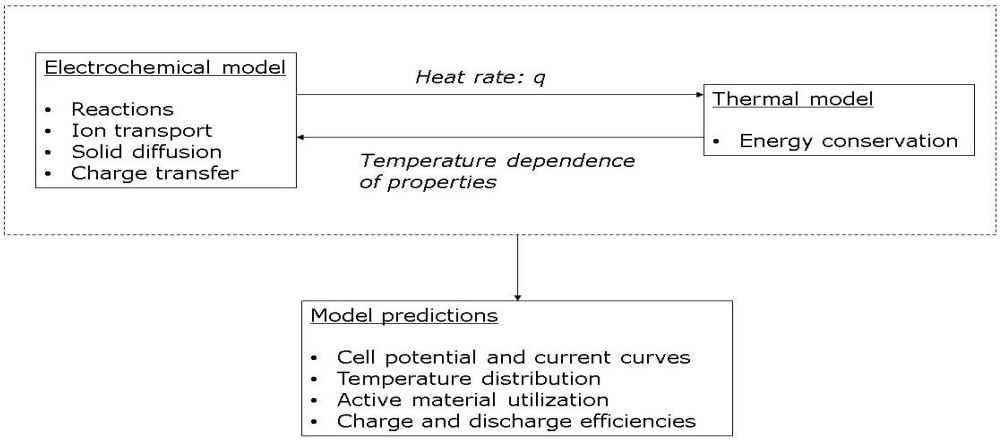

Figure 18. Coupled modelling approach replicated from (W. B. Gu and Wang 2000a).

3.6. Model validation

In this section, the present model is compared against the solutions developed by (B. Paxton and Newman 1997). Further comparison is established against available data from (COMSOL 2011).

Unless otherwise mentioned, all studies were carried out under isothermal conditions.

3.6.1. KOH profile

Figures 19 and 120 show that the KOH profile is fully developed in about 6 minutes after discharging, the difference across the cell is less than 1M. Comparison with (B. Paxton and Newman 1997) work shows good agreement of the model developed. That slight difference of around 1M shows that concentration gradients are low and the cell experiences only minor concentration polarisation due to electrolyte transport limitations.

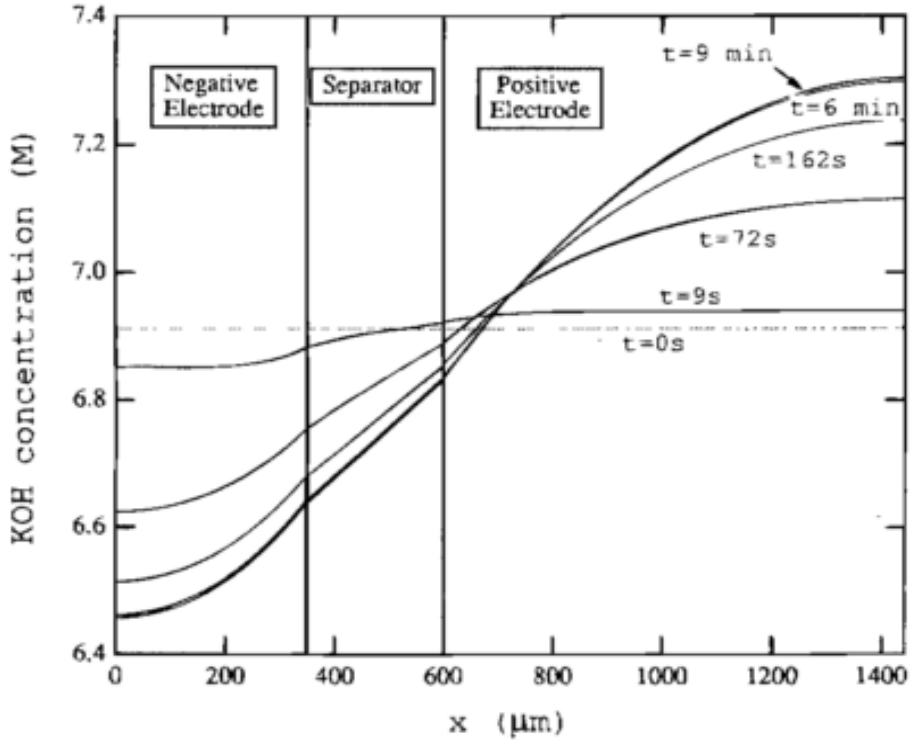

Figure 19. COH concentration profiles, from Paxton.

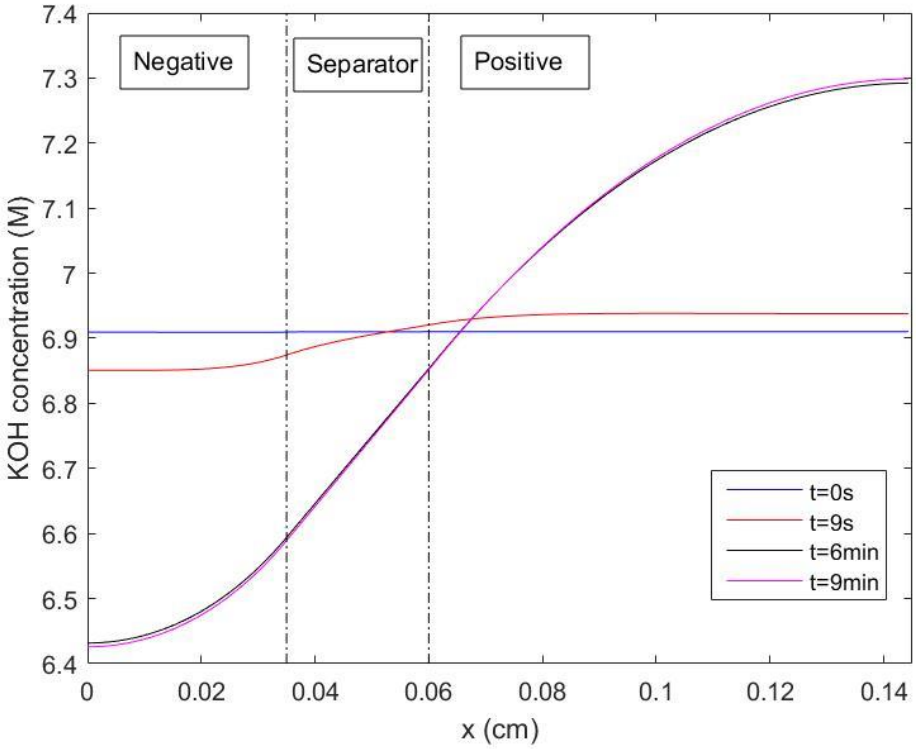

Figure 20. COH, concentration profiles. This work.

3.6.2. Discharge curves

The shape of the discharge curves (figures 21 to 24) is dominated by the near-Nernstian behaviour of the nickel oxide electrode thermodynamics (B. Paxton and Newman 1997). Though the shape of the discharge curves is as expected, at the first-time step and at near end of discharge there is disagreement with the work of Paxton. Also, available data in Comsol Multiphysics (COMSOL 2011) show discrepancy at start of discharge value.

The theoretical larger values at the beginning of discharge are due to the non-inclusion of side reactions in the model. Moreover, Paxton included a solid phase hydrogen concentration potential term in his work that this work and Comsol do not contemplate.

The differences are also due to the approach taken of the solid-phase diffusion of the spherical particles explained in chapter 1 and 3. The discharge capacity decreases slightly at 1C, though, at near end of discharge the works compared show different shape, they all reach a cut off voltage of 1V at 95% of discharge, concretely at 98% in this work. In lower discharge rates C/2 and C/6 the difference is larger, it cannot be determined whether it is due to the reasons mentioned above or a bad characterisation of the positive electrode.

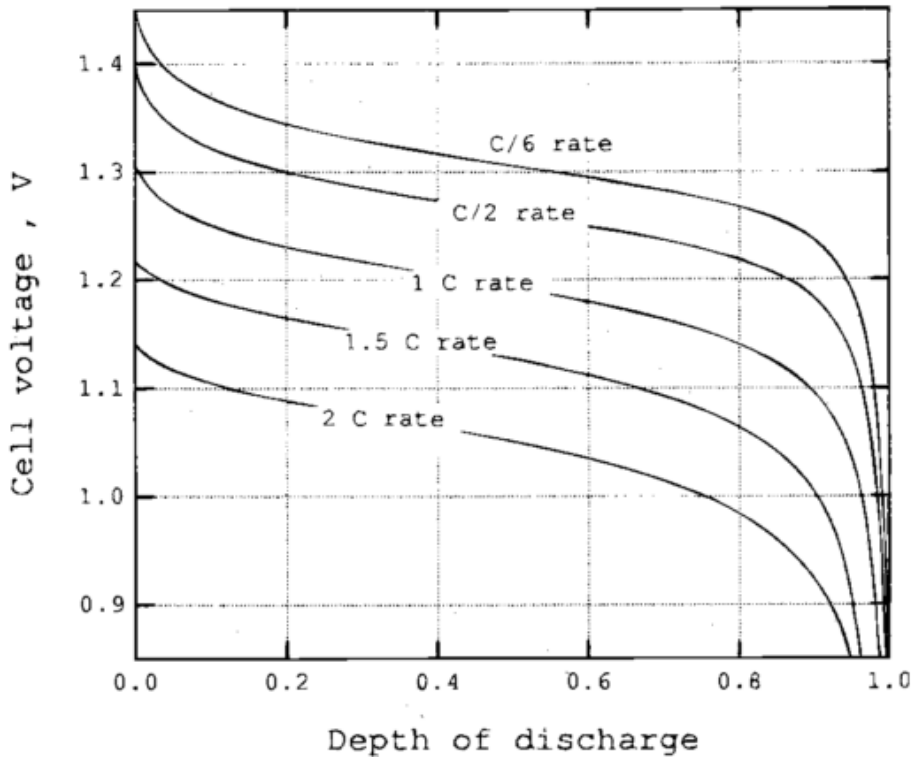

Figure 21. Discharge curves from Paxton

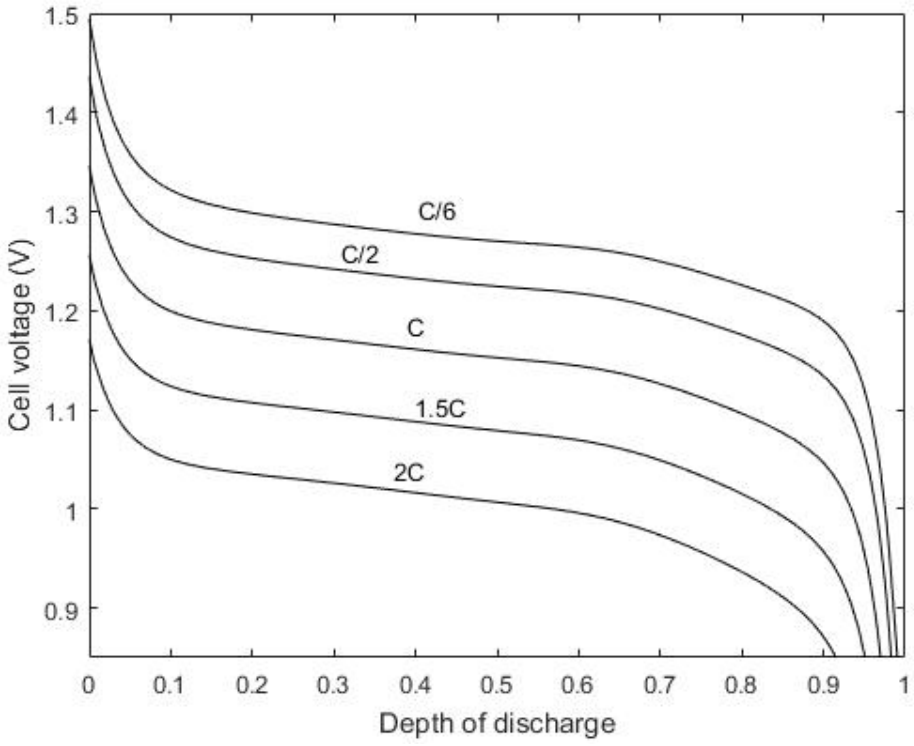

Figure 22. Discharge curves. This work.

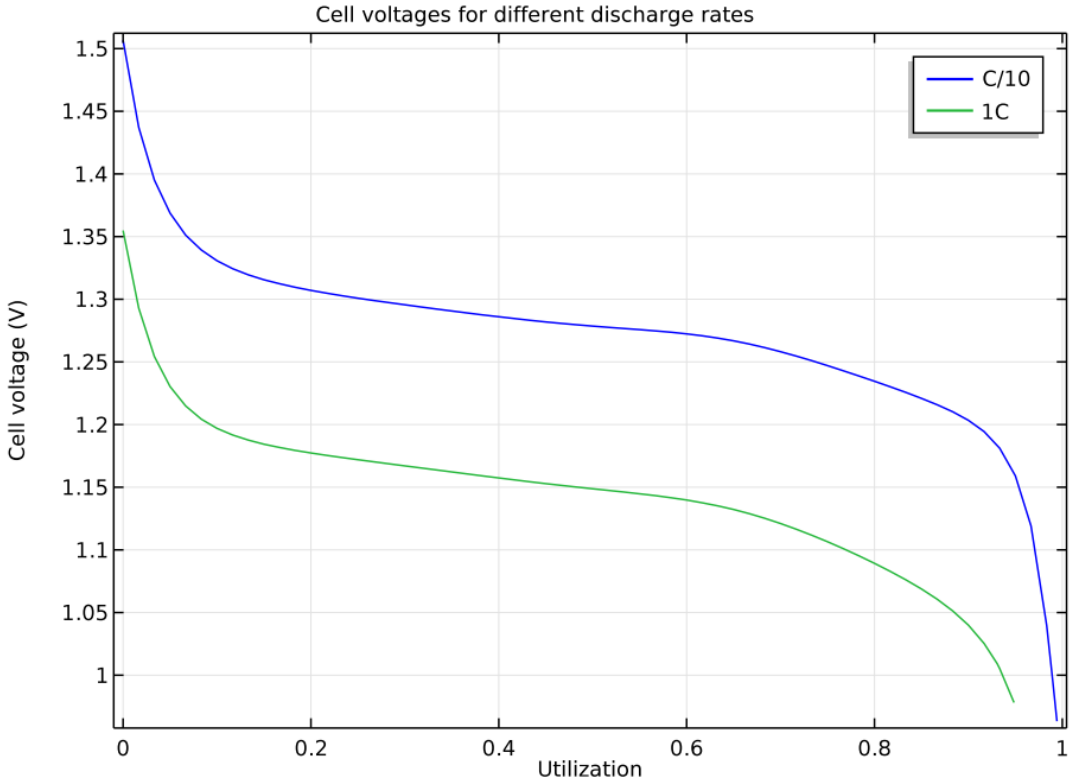

Figure 23. Discharge curves, from Comsol.

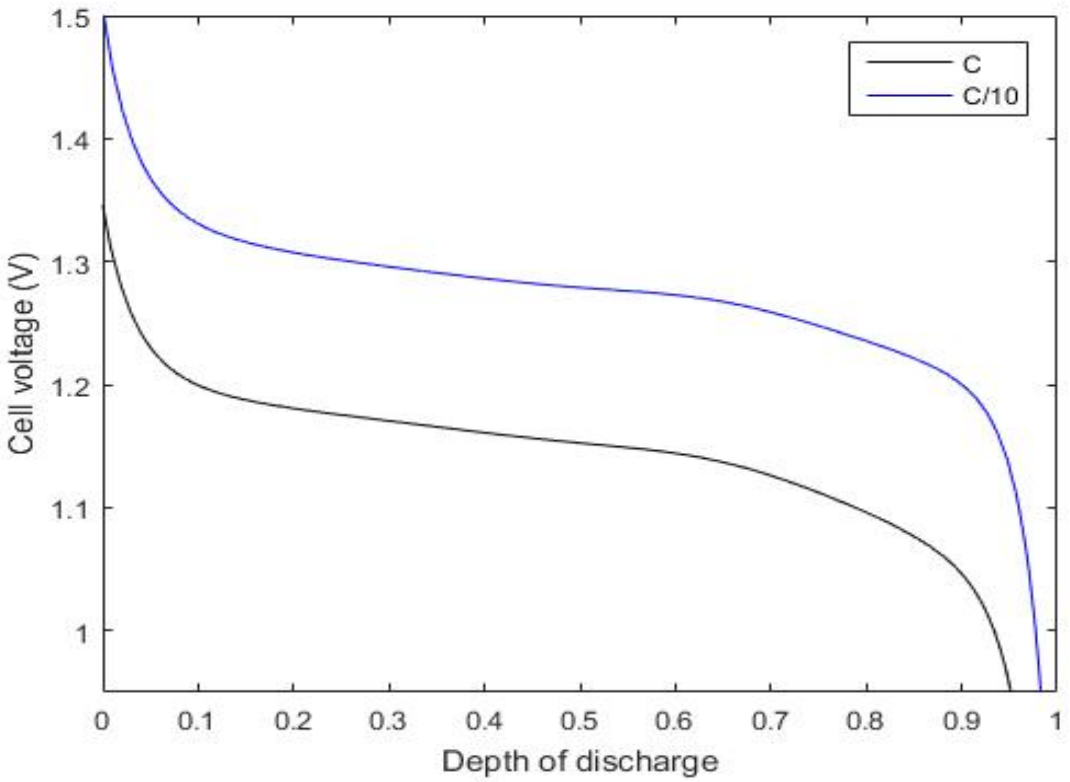

Figure 24. Discharge curves. This work.

3.6.3. Losses

Figures 25 and 26 presents voltage losses at 600s of simulation comparing (COMSOL 2011) and this model, for that comparison the plots are displayed within a similar range of values, that is, the electrolyte potential is plotted with a bias of 0.91V. The largest contributor is the overpotential in the positive electrode at 1C discharge, that was expected as the battery is positively limited. Notwithstanding, there are evident differences with the current work.

The overpotential is lower in comparison, though the shape is the same. The electrolyte potential line is not as smoother in Comsol, most likely because this program makes a mesh for each region of the cell, whereas, this work uses a mesh for the whole cell. This should be meaningless, the results obtained ought to be the same regardless of defining a mesh for each region or not, nevertheless, it is impossible to compare further in this matter as there is no available information to do so.

The potential losses across the phase interface across the cell is illustrated in figures 27 and 28. The largest contribution is in the nickel electrode, that is because the cell is positively limited, therefore, kinetics of the nickel electrode needs a closer look(B. Paxton and Newman 1997). All curves start near to zero and become slightly larger during the discharge.

The results are compared with those of (B. Paxton and Newman 1997), because they made a different mathematical approach of the problem (they took the same reference electrode in each electrode), in this work has been needed to introduce an U value so that the results are the same. The curves have similar shape, however, a more pronounced peak is obtained in this work, whereas Paxton and Newman obtained a smoother slope.

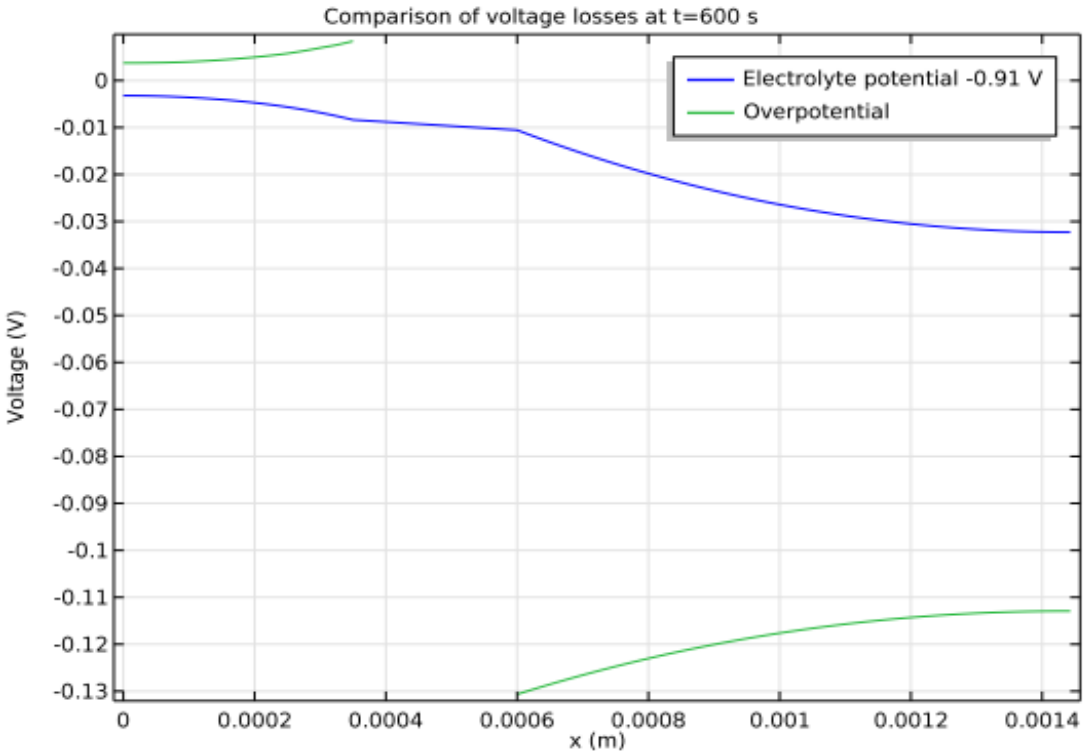

Figure 25. Voltage losses, from Comsol.

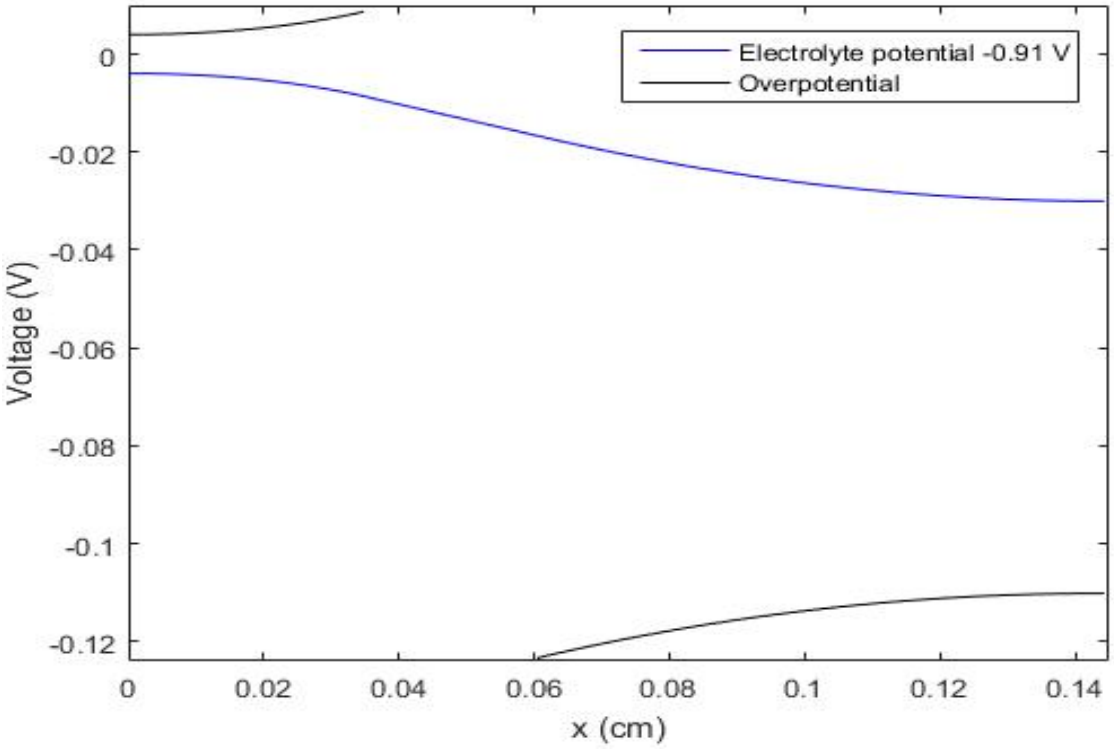

Figure 26. Voltage losses. This work.

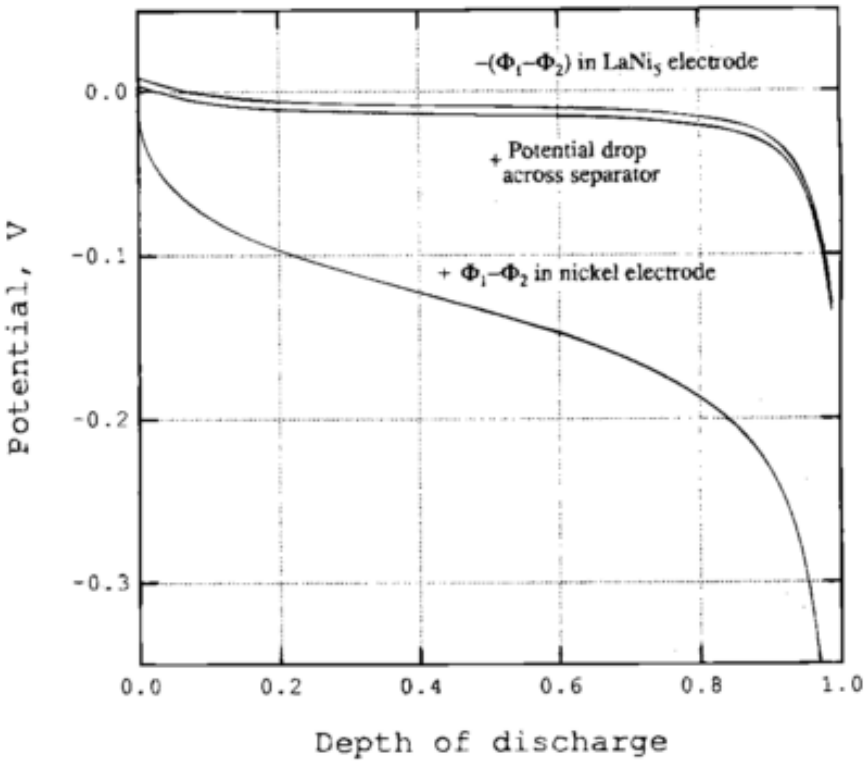

Figure 27. Potential losses, from Paxton.

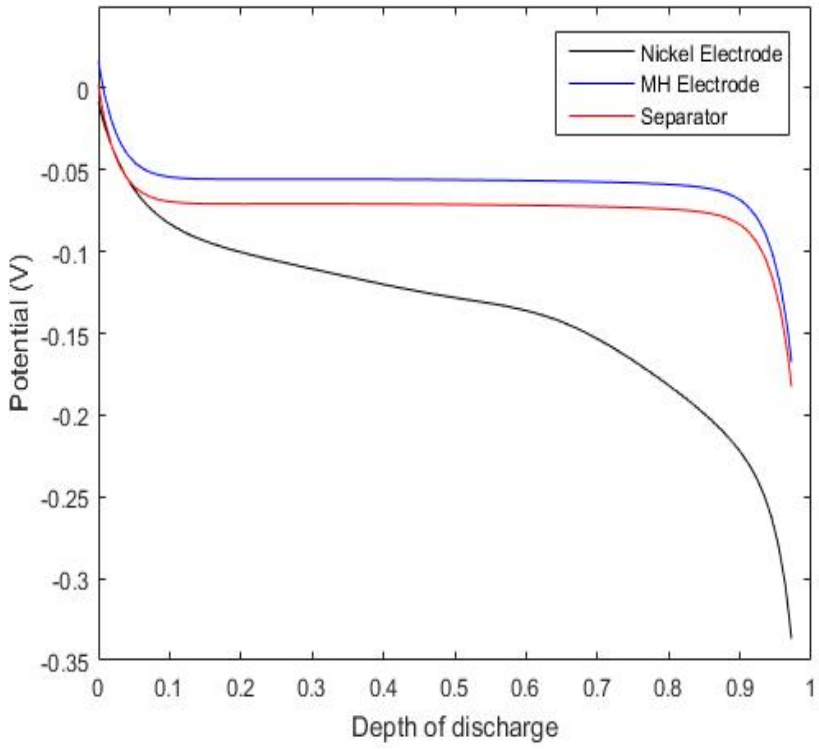

Figure 28. Potential losses. From this work.

3.6.4. Solid species concentration

The solid phase concentration was expected not to suit compared works because of the approach taken. Figures 29 and 30, show the comparison between this model and the data available from (COMSOL 2011).

It is utterly important to recall the explanation in Chapter 1 regarding the solid phase approach. In this work, as explained in Chapter 3, a single particle model has been chosen (DeVidts, Delgado, and White 1995) and an interfacial balance has been taken to account for the concentration variation alongside the electrodes (Gu, W. B. Wang 1998); however, the best approach would be to solve the solid phase concentration equation at each volume control of the discretisation.

The difference between the simplification made in this work and the proper resolution of the intercalation species is the slight downward curve appreciated at 600s and more pronounced at 3000s. It cannot be determined, however, when that difference starts being significant, it seems that until 600s the results show good agreement with the concentration over 80000 mol/m^3 in the negative electrode and below 10000 mol/m^3 in the positive. Alas due to the lack of information regarding this variable, it is impossible to discuss the matter further.

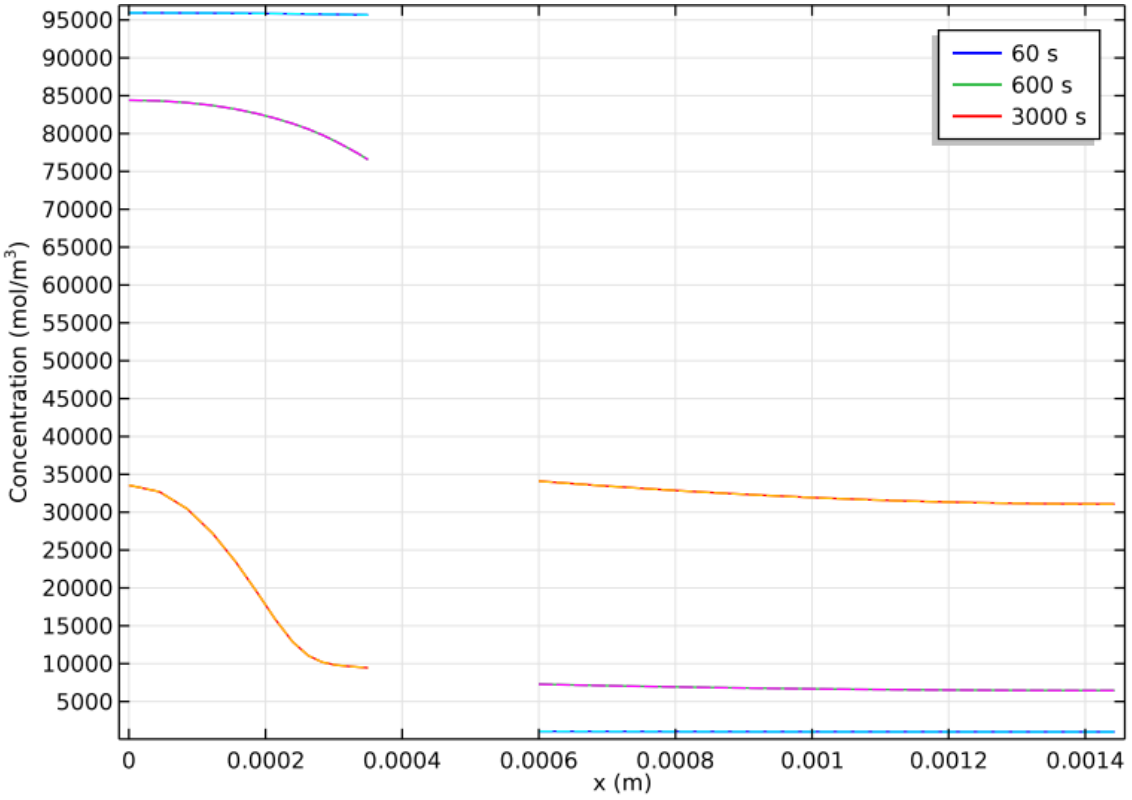

Figure 29. Solid phase concentration, from Comsol.

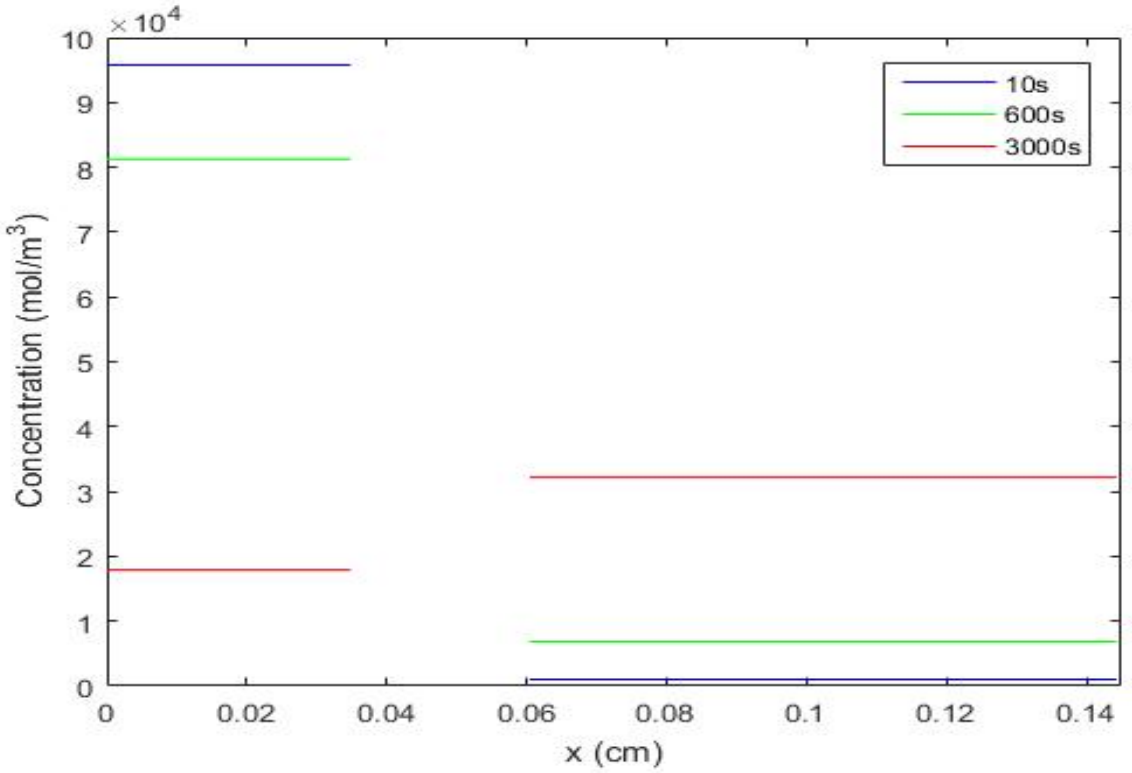

Figure 30. Solid phase concentration. This work.

3.6.5. Temperature variation

In literature the temperature is usually plotted versus the SOC (State of Charge). Few studies are devoted to heat evolution and none has been found on entropy evolution to reproduce temperature results it was necessary to redesign the problem to focus on charge. Though, the process is the inverse of discharge, results are not the inverse. During charge, it has been assumed that the nickel electrode's solid-phase concentration is almost maximum and nearly depleted for the other electrode. The boundary condition for discharge in the solid-phase potential with a negative sign account for charge.

Likewise, thermal models giving enough parameters to compare are scarce. The temperature profiles were to be compared with those of (W. B. Gu and Wang 2000a). However, they included oxygen evolution, pressure variation and a term accounting for the metal hydride enthalpy of formation. Furthermore, the information provided in those works is not clearly described and, therefore, it has been not possible to reproduce it.

To examine results first is needed to know the order of magnitude of the convective heat transfer coefficient h. Provided that no cooling system exists, the most likely heat transport medium is air, whose convective heat coefficient ranges from 0.6 to 3 W/m^2K for free-convection and from 3 to 30 for forced convection (Pals and Newman 1995). Adiabatic condition is also assessed, in case there was no heat removal, e.g., temperature control failure (Pals and Newman 1995).

Reported results state that under adiabatic conditions, temperature raise up to 80ºC, under forced convection (25 W/m^2K) temperature rises 5ºC from the initial temperature and under free convection (5 W/m^2K) it raises about 10ºC (W. B. Gu and Wang 2000a).

In non-isothermal simulations, temperatures profiles under different convection coefficients are shown in figure 31. Though the increase above mentioned is fulfilled, it cannot be compared to other author's works since they included side reactions and pressure variation (W. B. Gu and Wang 2000a) or modelled lithium batteries (W. B. Gu and Wang 2000b). Therefore, to make the comparison, the model should be modified, incorporating the mathematical simulation of such reactions, which does not correspond to the objectives of the present work. Temperature curves in adiabatic conditions for different discharge rates are examined in figure 32. As expected higher rates lead to higher temperature (Pals and Newman 1995).

There is also a maximum temperature to avoid (horizontal dashed line in figure 32) for safety reasons, to avoid melting or decomposition. Consequently, figure 32 is useful to predict the maximum discharge rate to avoid that critical temperature (Pals and Newman 1995). For an aqueous cell the safety limit is set to 80ºC showing the necessity of thermal management for Ni-MH batteries (W. B. Gu and Wang 2000a).

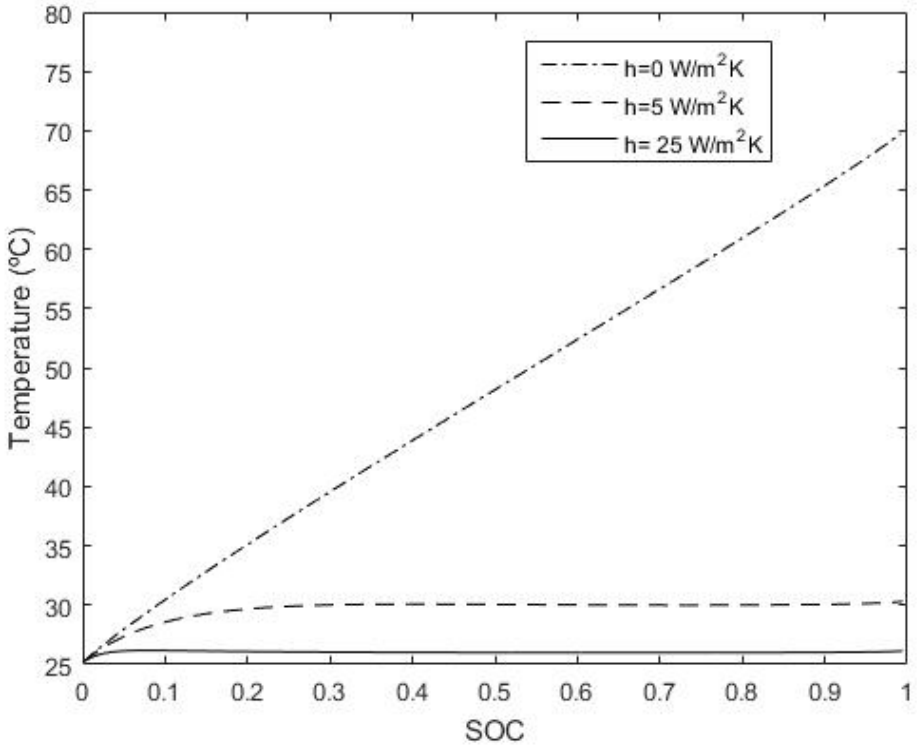

Figure 31. Temperature profiles during 1C charge for different convection coefficients.

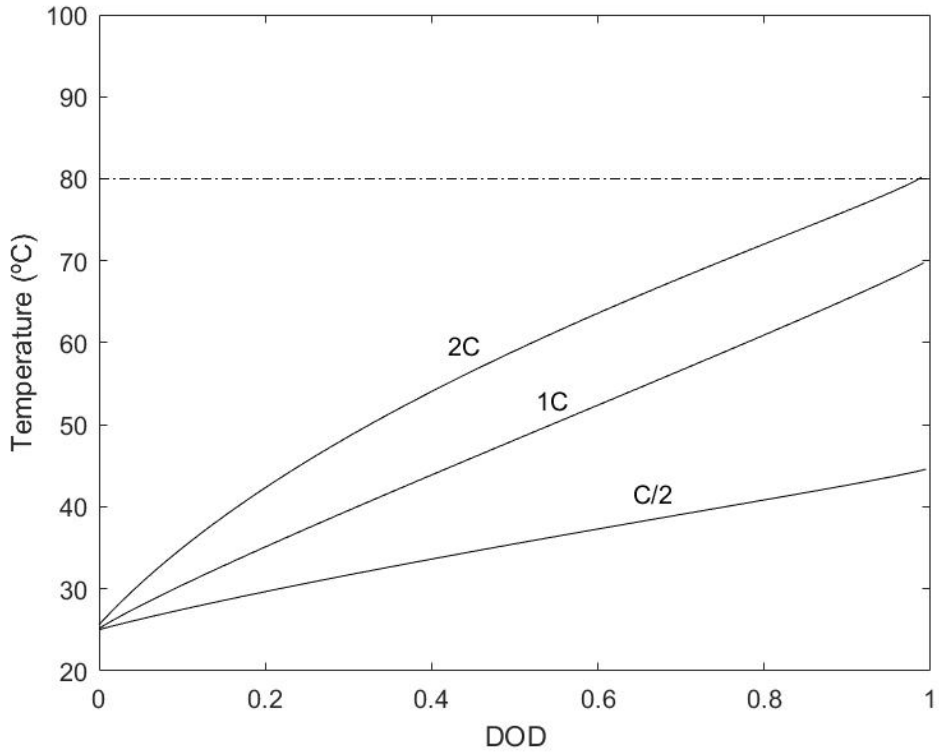

Figure 32. Temperature during 1C charge on adiabatic conditions.

3.7. Model analysis

3.7.1. Temperature analysis

Higher temperatures enhances the battery behaviour (W. B. Gu and Wang 2000b), enhancing species transport and boosting kinetics. Figures 33 and 34 show that on adiabatic conditions (higher temperature increase) the cell voltage reaches the cut-off voltage with a greater DOD for both 1C and 2C cases.

Before further assessment is made, it is important to model isothermally at different temperatures. According to (Pals and Newman 1995), if a battery exhibits a strong dependence on the temperature of operation, the thermal-management must be precise, otherwise, it does not urge to be so precise. Secondly, they state that the optimum temperature of operation defined from isothermal data must be inside a range wide enough to allow simple temperature control system but narrow enough not to adversely affect performance.

Isothermal discharge curves for different temperatures as a function of positive electrode utilization are shown in Figure 35. As expected higher temperatures yield better performance of the cell. Such improvement is noteworthy from temperatures of 45ºC and higher, and is likely due to the increase of the cell's properties (diffusion, ionic conductivity) with temperature.

The utilization is similar to a local state of charge or discharge, it tells how depleted or charged an electrode is. It is defined as the concentration of the solid phase divided by the maximum concentration available:

$$Utilization = c_s/c_{s,max} \qquad 3.2.22$$

At 25ºC, 97% of the active material is used and at 45ºC 99% is used, temperatures close to 80ºC are discarded to avoid thermal complications (W. B. Gu and Wang 2000a). The end of discharge concentration profiles on different temperatures are depicted in figure 36, the higher the temperature the more uniform the concentration is. It can be also observed that on this range of temperatures, the cell is not diffusion limited.

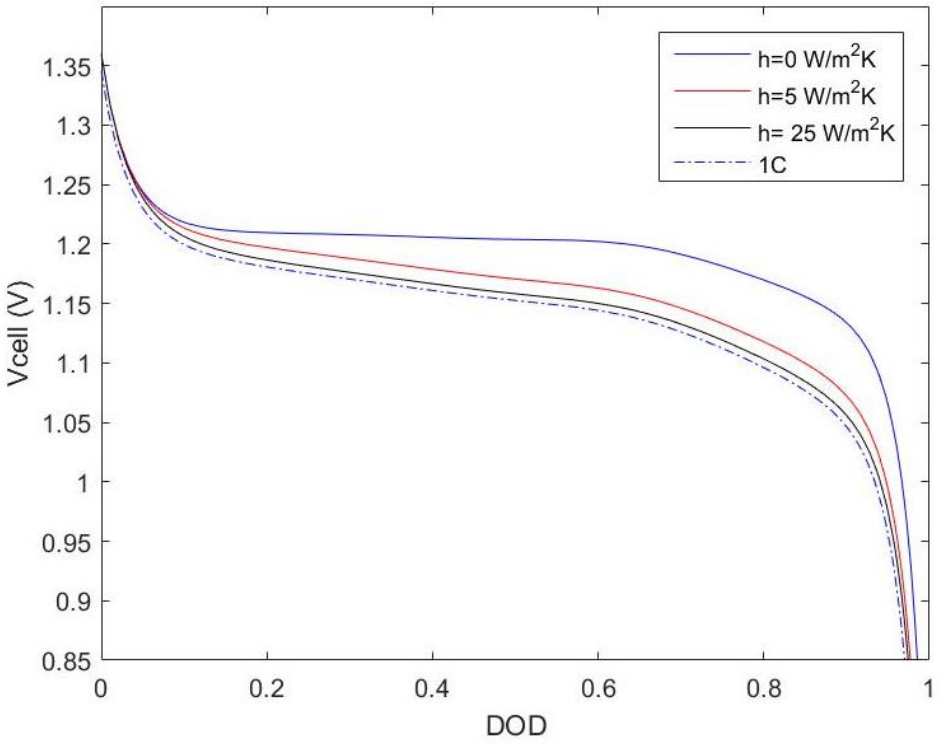

Figure 33. Discharge curves on different convective coefficients for 1C discharge rate.

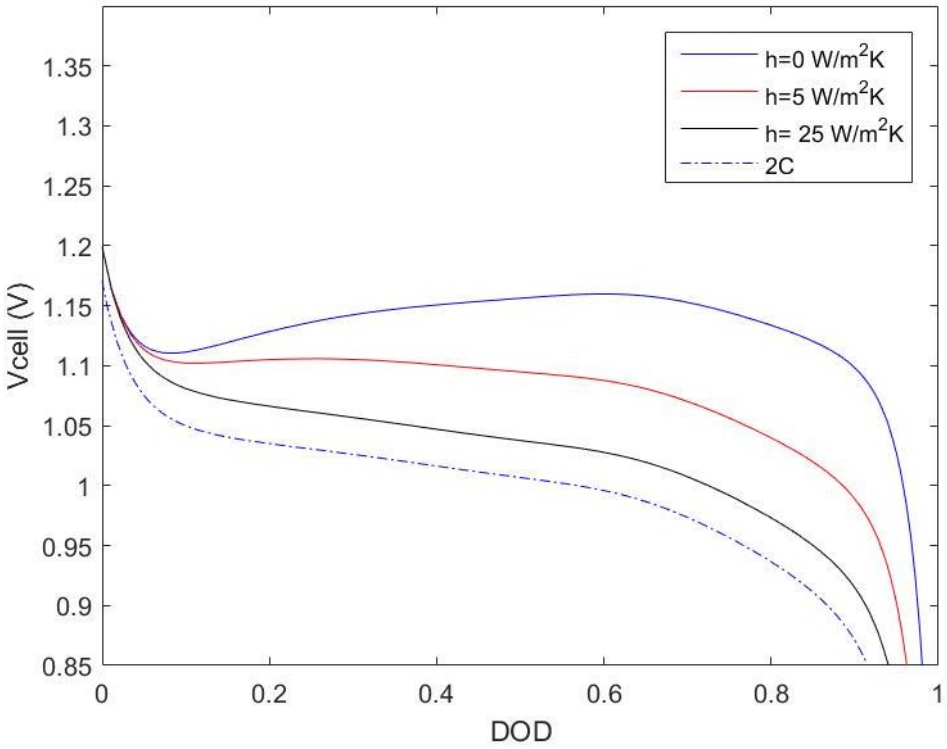

Figure34. Discharge curves on different convective coefficients for 2C discharge rate.

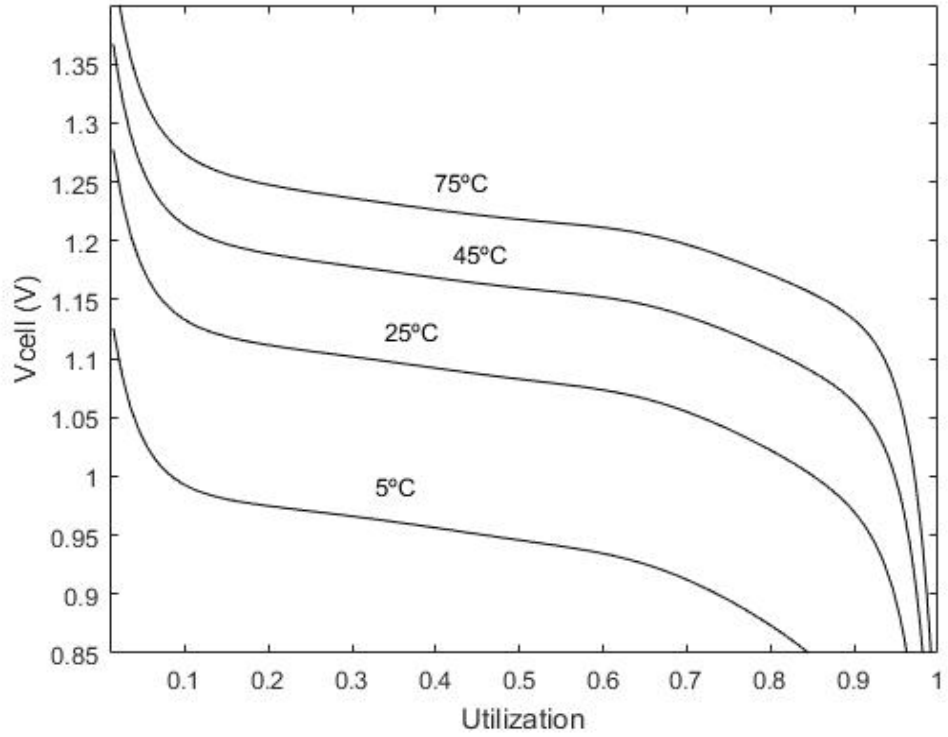

Figure 35. Isothermal discharge curves for several temperatures at 1,5C discharge rate.

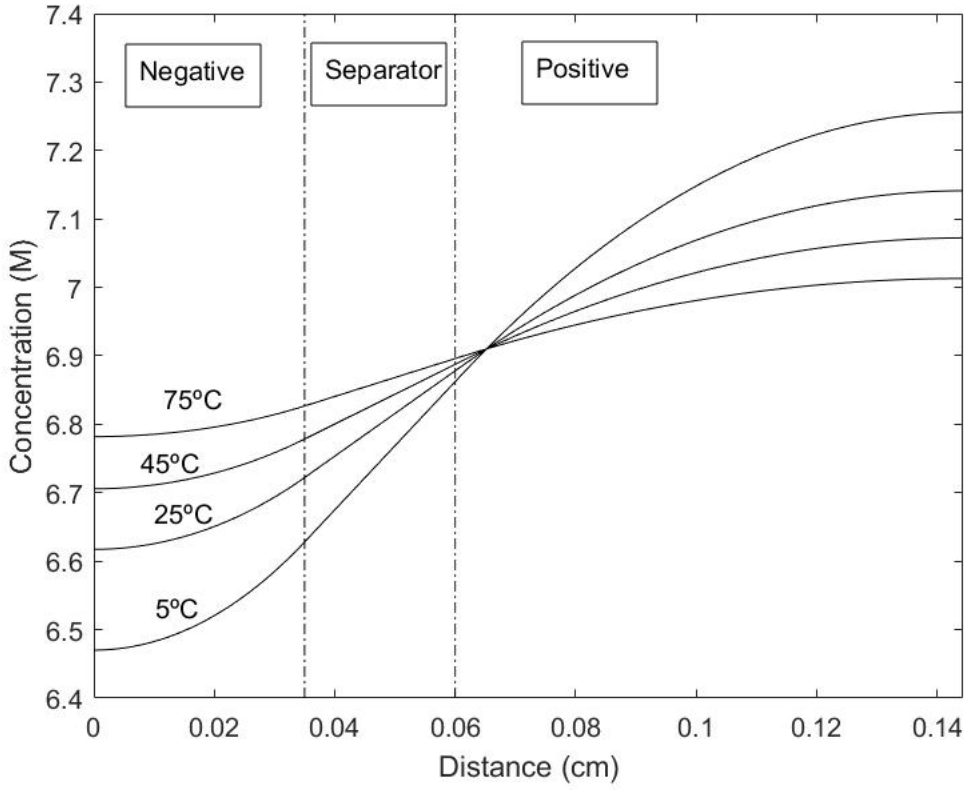

Figure 36. End of discharge concentration profiles at 1,5C discharge rate for several temperatures.

3.7.2. Charge and discharge cycles

A charge and discharge cycle is the process of charging and discharging a rechargeable battery. This term is usually used to specify a battery's expected life, i.e., how many times a battery can undergo charge and discharge processes until failure.

The following modifications are needed in order to compute charge and discharge cycles:

- Compute a cycle vector, from one to *n* cycles.
- Since simulations are time based, the final computational time shall be the theoretical time needed for fully discharge multiplied for the number of cycles.
- An *if* condition is needed at the end of time loop. When the potential reaches the upper or lower limit of cut-off voltage, the input current density changes its sign, i.e., $I = -I$ at the specified conditions.

The loss of capacity in Ni-MH batteries that limit its cycle life, is mainly due to side reactions in the negative electrode (Rahn and Wang 2012). Thus, the need to implement charge and discharge cycles in the model in order to study the degradation through time.

One and a half discharge/charge cycles are illustrated in figure 37, comprising: a full discharge, a full charge, another full discharge and half charge.

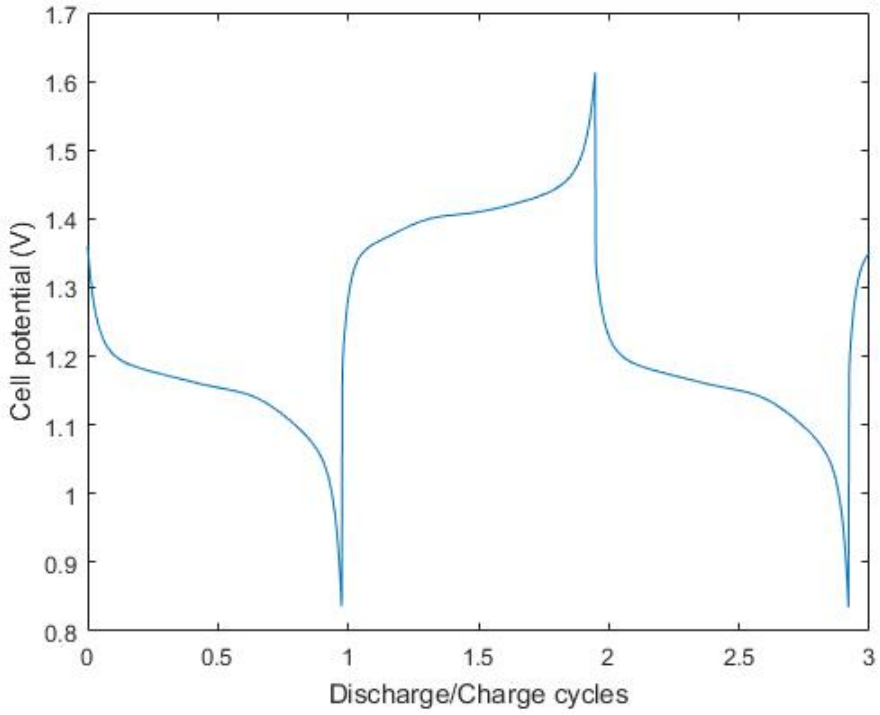

Figure 37. One and a half discharge/charge cycles.

3.7.3. Entropy assessment

To assess degradation via entropy key parameters were changed. Particularly, the dependence of electrical conductivity, porosity and maximum concentration of solid phase species were evaluated.

Discharge/charge cycles were carried out by setting a cut off voltage of 1V for discharge and 1,5 V for charge. The entropy was evaluated at half of discharge or charge, i.e., $\dot{S}$ at $DOD\ or\ SOC = 0{,}5$. Figures 36 shows the results of entropy variation on 71 cycles, a variation of 1% of the abovementioned parameters per cycle was assumed. Though, it is yet unclear the trend of degradation of these parameters. A linear approximation was chosen in the first place. After each cycle, the temperature is reset at 25ºC.

Three scenarios were contemplated. First, two simulations were carried out modifying only porosity and conductivity in the first one and the maximum concentration in the second. The last simulation varied all the parameters. This is illustrated in figure 38.

A zoom of figure 38 is shown in figure 39. It is concluded that capacity of cyclable species has the largest effect on degradation. Albeit a narrow decrease when varying the maximum concentration of solid phase concentration, all studies show an increase of entropy. This is in accordance with the thermodynamic concept that irreversible processes always have a positive generation rate.

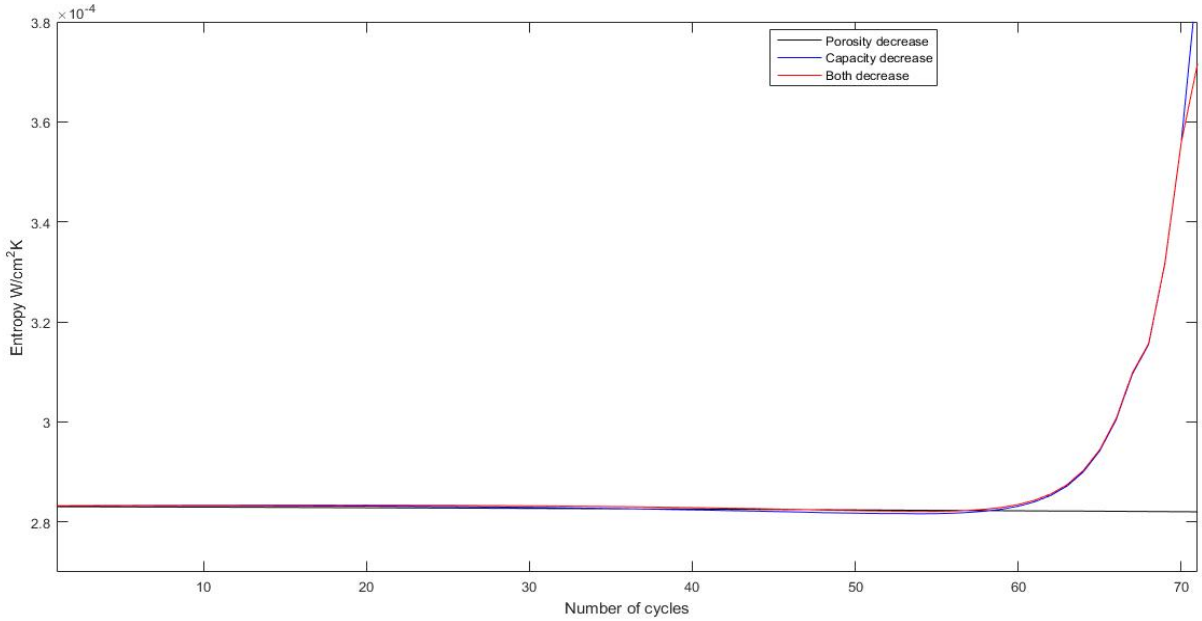

Figure 38. Entropy variation during discharge/charge cycles.

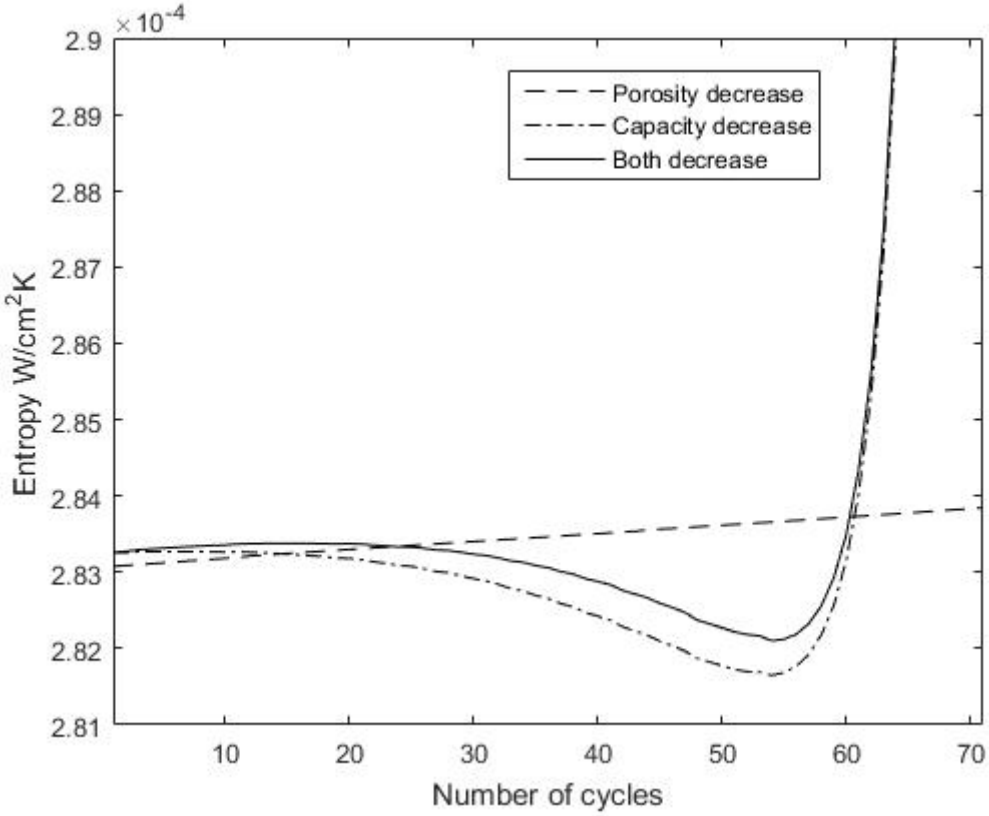

Figure 39. Zoom of figure 36.

To further assess the variation of entropy, it is essential to evaluate the entropy during discharge. To do this, simulations employing higher and lower values of porosity and conductivity were conducted. These are resumed in table 4.

Table 4. Modified parameters to assess entropy.

Variable	Initial value	Decreased Value	Increased Value
Electrical conductivity (σ)	1000 (negative electrode)	100	10000
	28 (positive electrode)	1	1000
Negative electrode porosity	0,481	0,2	0,6
Separator porosity	0,5	0,25	0,75
Positive electrode porosity	0,5	0,25	0,75

Figure 40 show the entropy rate on 1C discharge modifying the separator's porosity. The separator porosity has practically no effects on the entropy. This is because there is no reaction on the separator and, thus, no overvoltage.

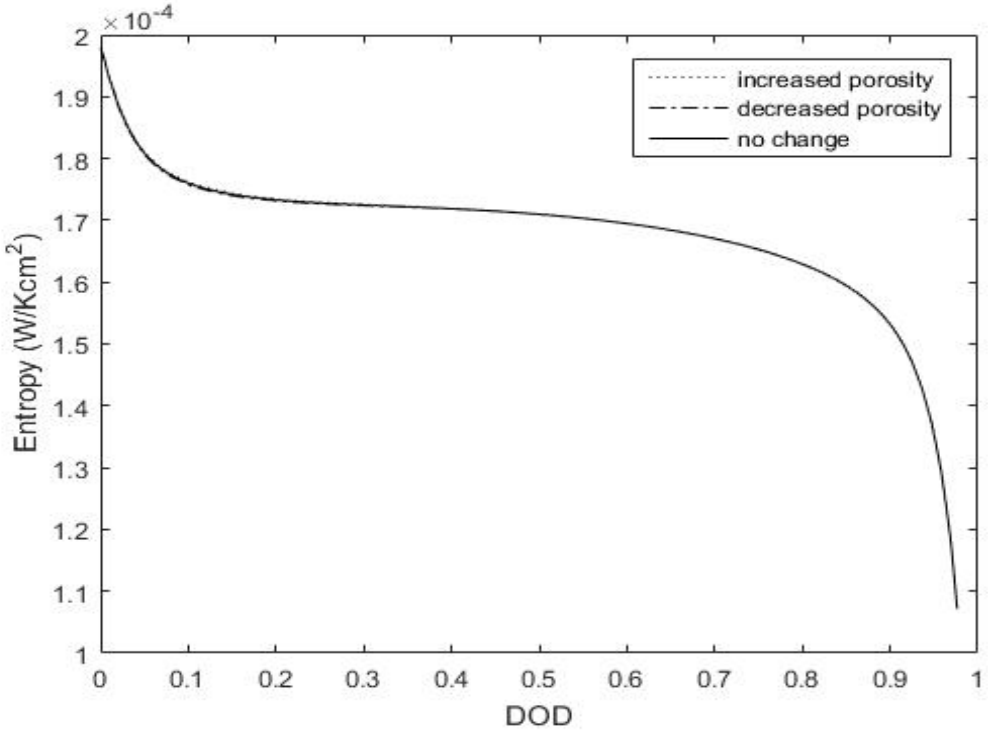

Figure 40. Entropy rate modifying the separator porosity on 1C discharge.

The upper and lower values of electrical conductivity have been taken from (Rahn and Wang 2012). The simulations show that there is a slight variation using the decreased values, as illustrated in figure 41. These variation, is due to the decreased value in the positive electrode, as illustrated in figure 42.

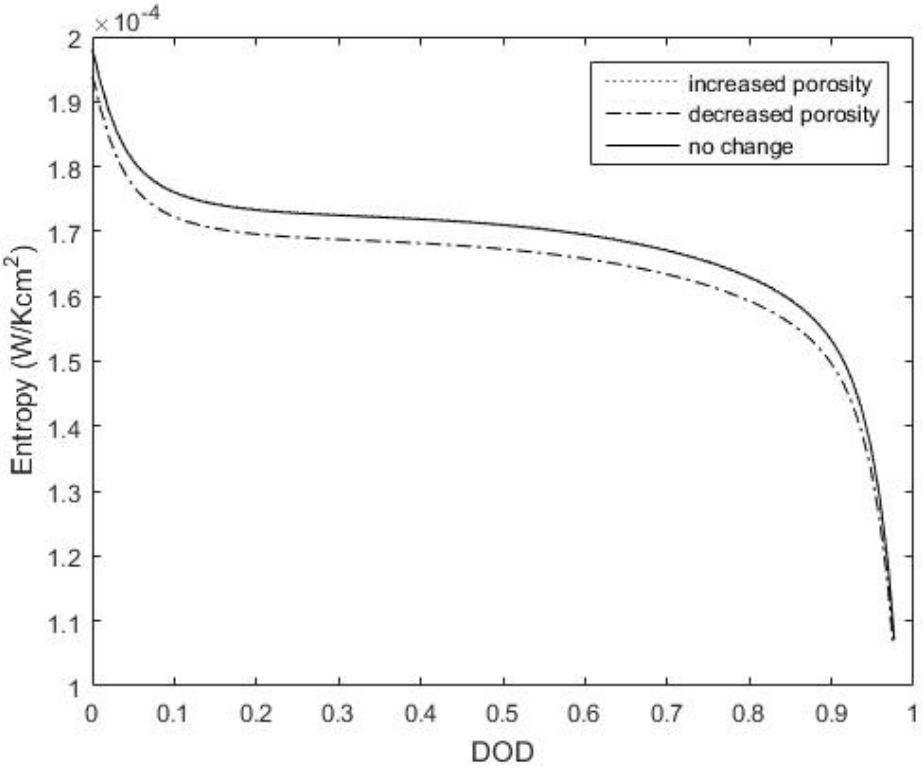

Figure 41. Entropy rate modifying the electrical conductivity on 1C discharge.

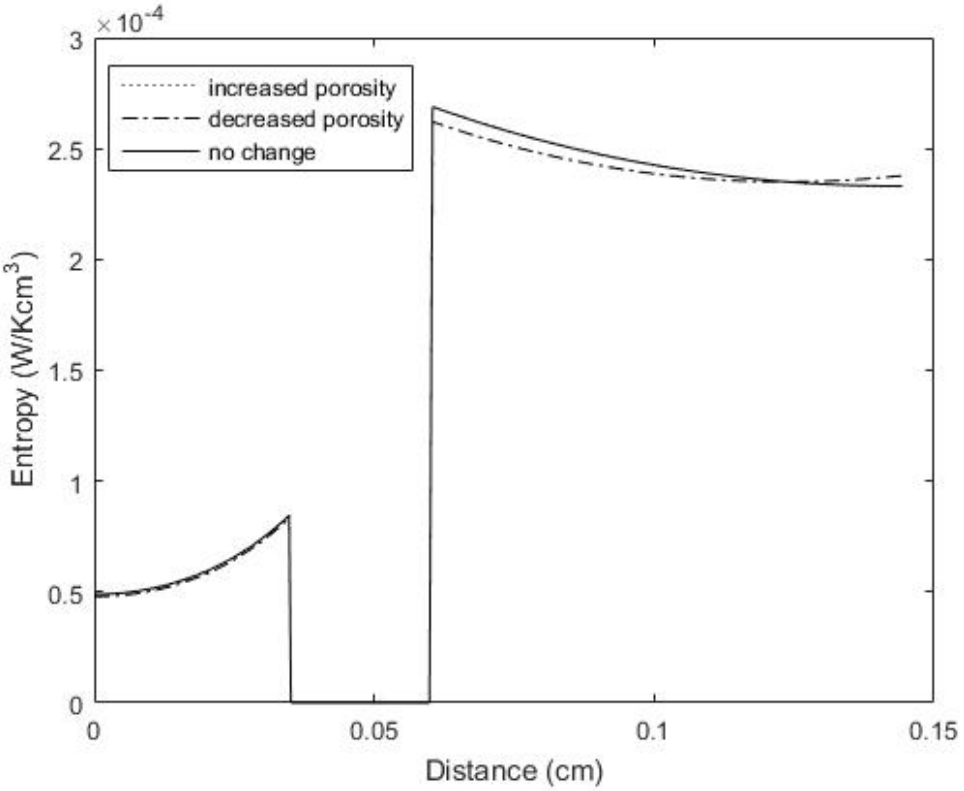

Figure 42. Local entropy rate modifying the electrical conductivity on 1C discharge.

A substantial decrease in the porosity of the metal hydride electrode affects drastically the battery's performance. Figure 43 show how the performance is diminished and a lower limit is reached at 0,4 DOD. In that case, the entropy decreases because the battery performance decreases as well. Recalling there is no literature regarding entropy simulations in batteries, the entropy's profile is controlled by the profile of the overpotential.

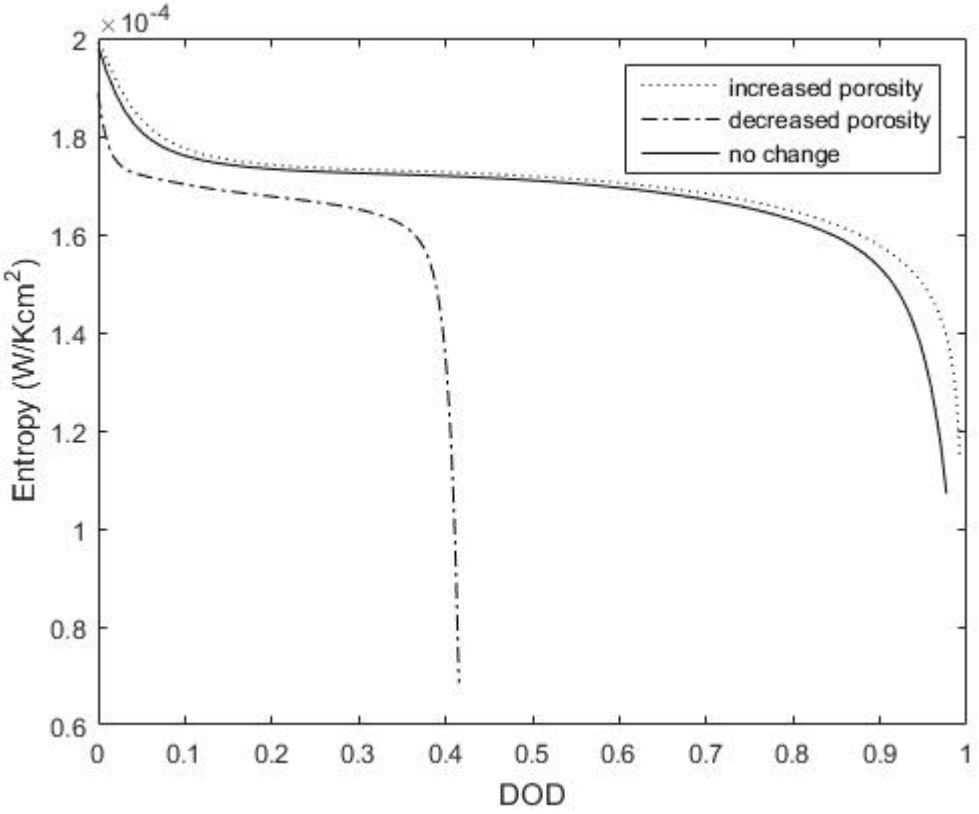

Figure 43. Entropy rate modifying the negative electrode porosity on 1C discharge.

Figures 44 and 45 show the local entropy rate. At 1C the difference is larger than at 3C. This has a noteworthy impact on the positive electrode showing how entwined are the variables that comprise the battery system. A change of potential or solid phase in one electrode as a result of a variation of porosity, impacts enormously on the other.

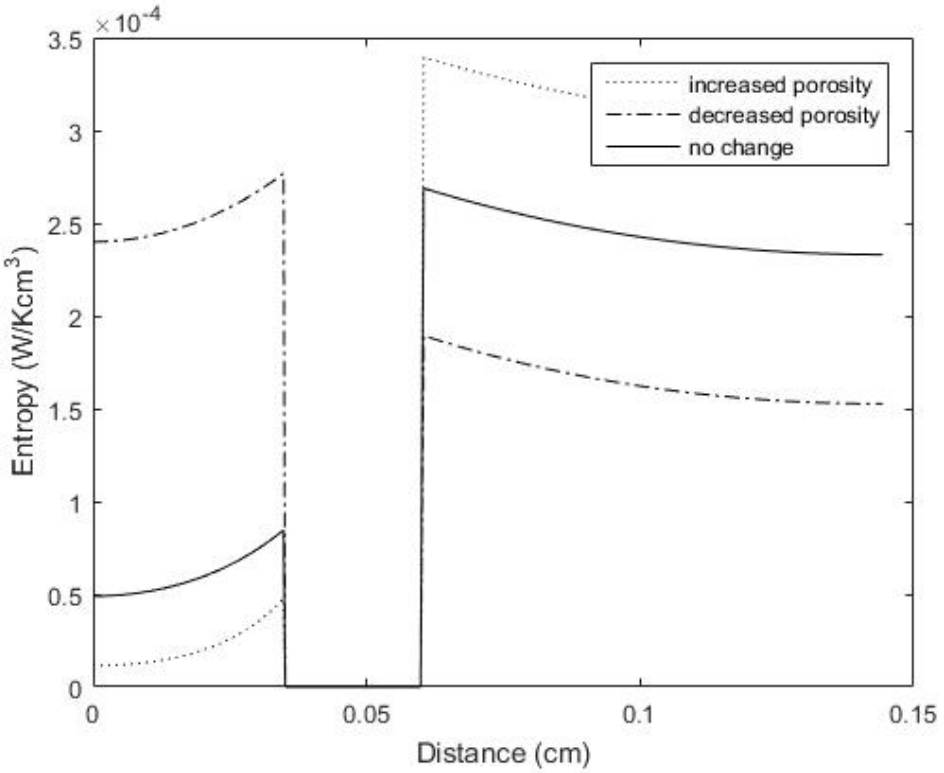

Figure 44. Local entropy rate modifying the negative electrode porosity on 1C discharge.

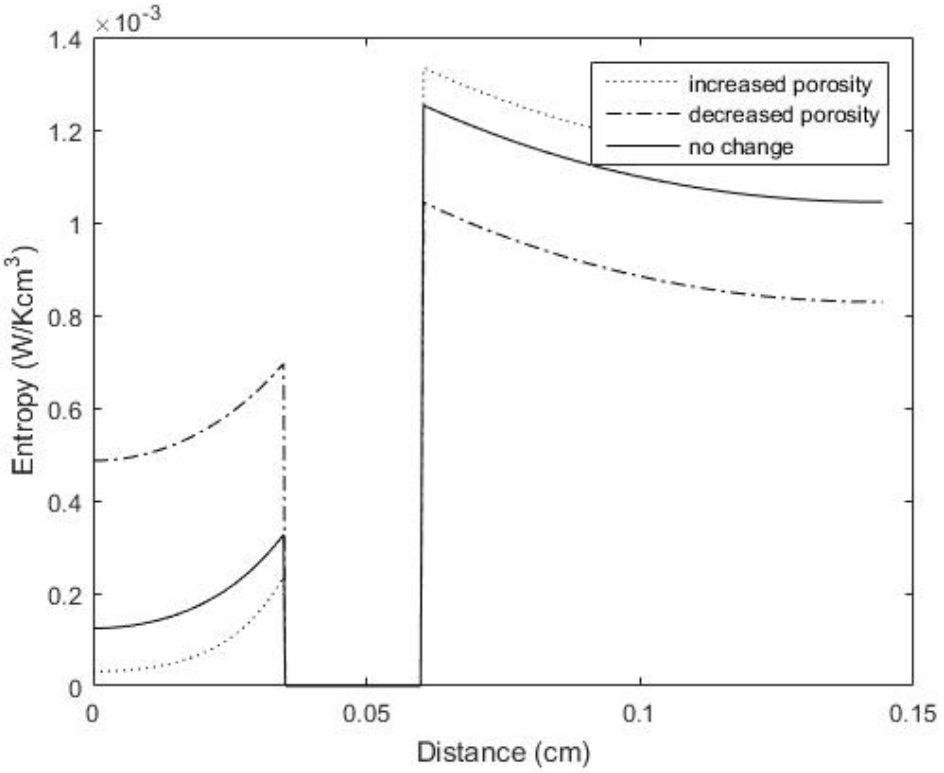

Figure 45. Local entropy rate modifying the negative electrode porosity on 3C discharge.

Likewise, decreasing the porosity in the nickel electrode had a huge impact on the functioning of the battery. Figure 46 shows the performance is diminished, reaching the lower limit at 0,5 DOD in a similar way as decreasing the negative electrode porosity. The performance is analogous to the negative porosity variation, though the impact on the other electrode is minor (figure 47).

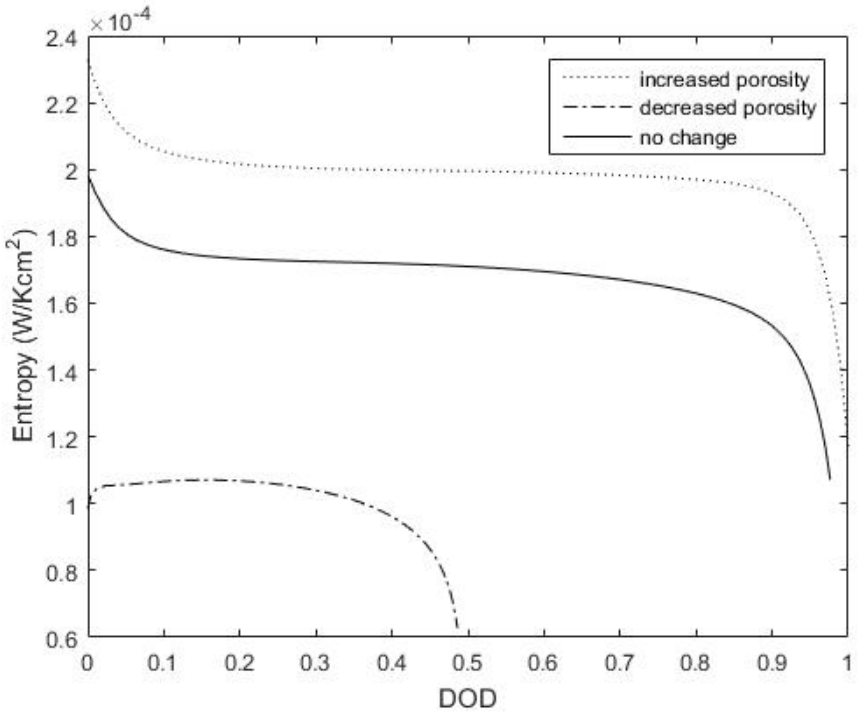

Figure 46. Entropy rate modifying the positive electrode porosity on 1C discharge.

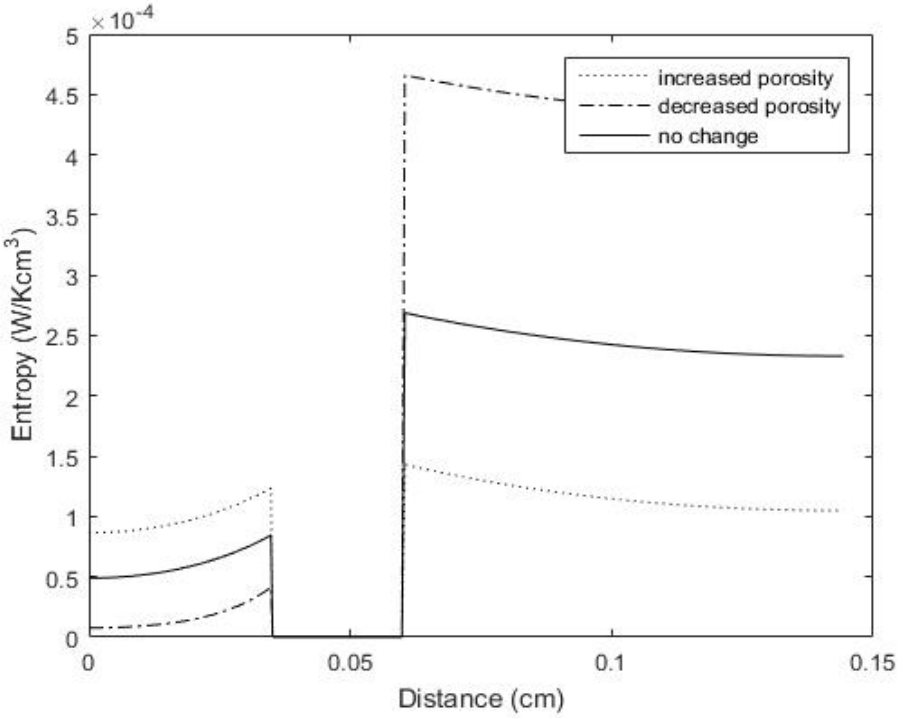

Figure 47. Local entropy rate modifying the positive electrode porosity on 1C discharge.

By modifying porosities and the electrical conductivity simultaneously (figure 48), the performance decays rapidly to a lower limit, reached at 0,4 DOD. This shows that, perhaps, the negative porosity has the largest impact on performance. The local entropy rate (figure 49) shows the effects on both electrodes. The entropy at both electrodes is lower than separately. This highlights the importance to study the whole system as already commented.

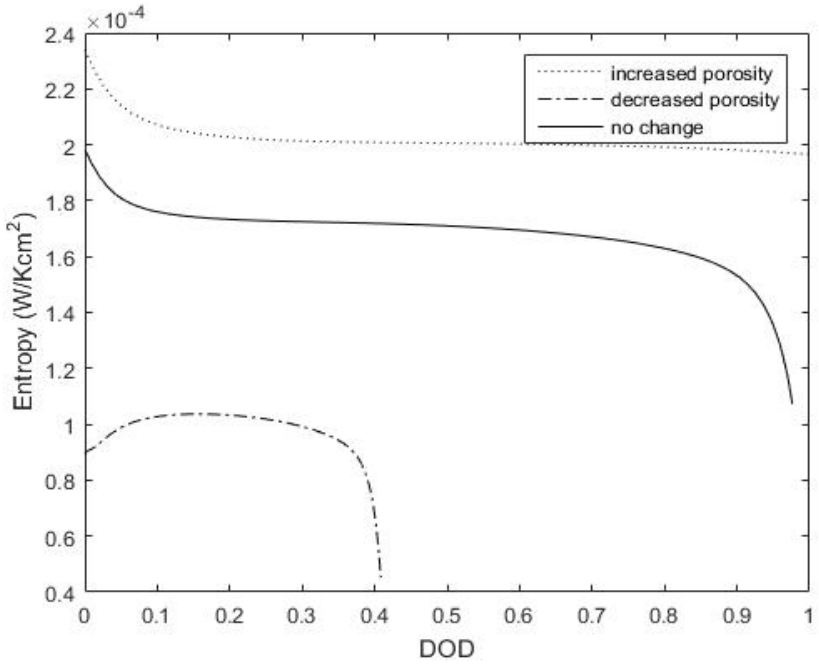

Figure 48. Entropy rate modifying all parameters on 1C discharge.

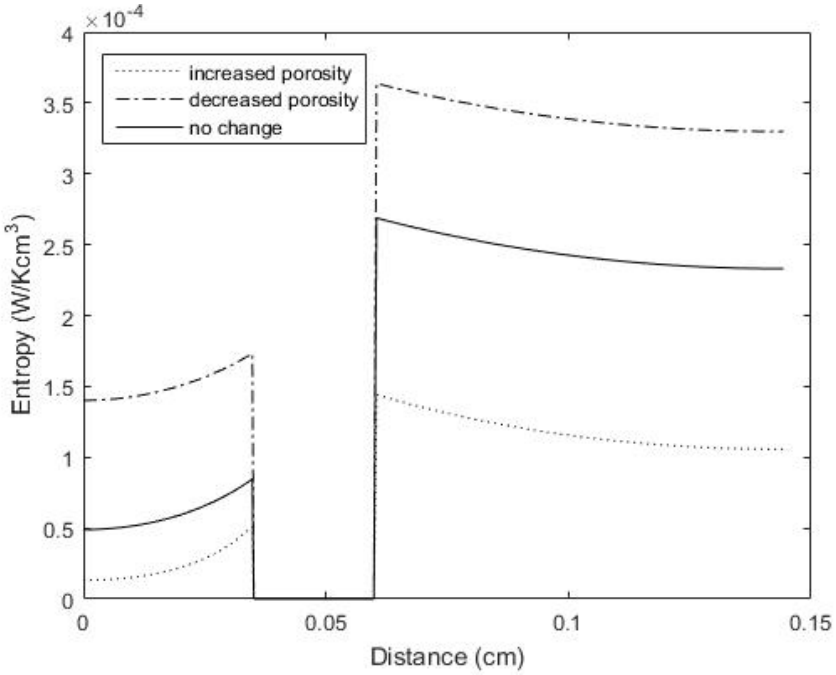

Figure 49. Local entropy rate modifying all parameters on 1C discharge.

3.7.4. Discussion

In the degradation cycles simulations, the capacity of the electrodes was diminished over a 51% at the end of life cycle, i.e., when due to the loss of capacity cycles could no longer be computed. Normally, the end of life is characterised by a drop of 50-80% of the initial capacity (Rahn and Wang 2012).

As the battery degrades, the internal resistance increases and, therefore, the irreversible heat rises. This leads to an increase in entropy, which has been seen in the previous section and to an increase of temperature. The latter is illustrated in figure 50. Conductivity, porosity and capacity decrease as a result of side reactions and in the electrodes.

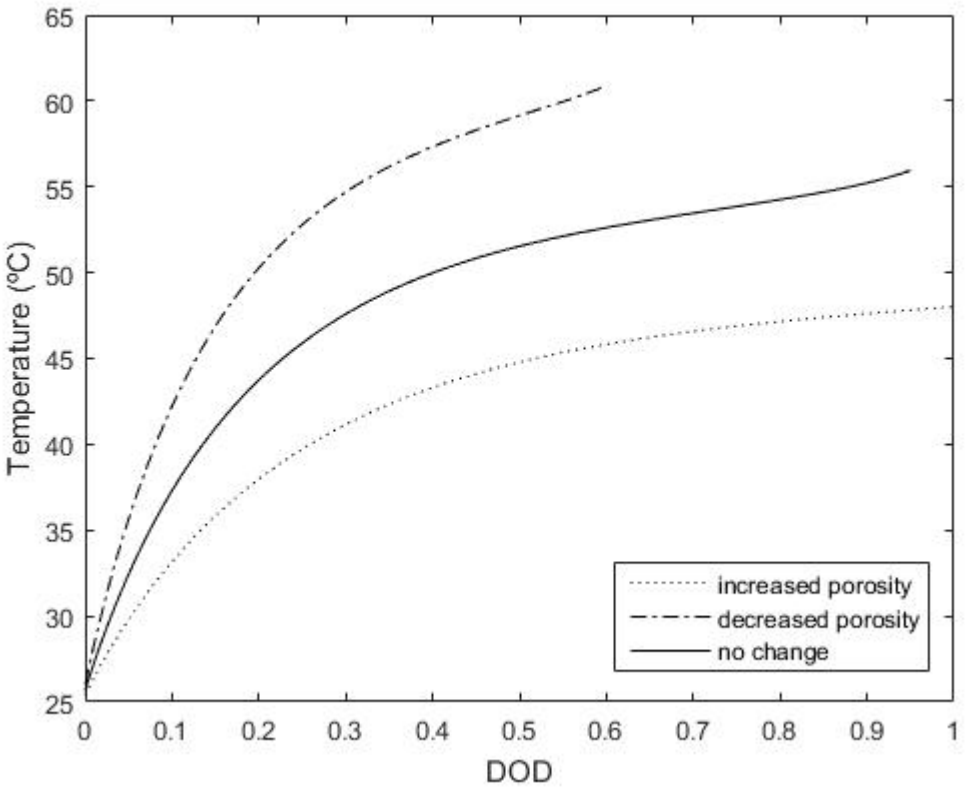

Figure 50. Temperature on 3C discharge with $h = 5\ W/cm^2K$.

Chapter 4 Environmental impact

As described in the introduction section, Ni-MH offer enhanced electrochemical properties, thus, they have been extensively used in electronic devices and have been considered as power source in electric vehicles. The recycling of this type of battery and the impact on the environment has, therefore, become an important issue. As well as the means to obtain the raw materials, Ni-MH batteries require the most energy for their production, about 90 MJ per kilogram of battery produced. If fossil fuels are used, the environmental impact expected is high (European Commission DG Environment New Alert Service 2012).

Ni-MH batteries are sealed to avoid leakage of electrolyte and the generated gas during discharge, making them safer and more acceptable towards the environment, since no hazardous materials are used, figure 51 illustrates its life cycle. They are, consequently, less toxic than cadmium or lead-acid batteries. Nickel and metal hydride are valued materials and research has been undertaken in order to recycle them (Zhang et al. 1998)(Wakui and Inoue 1998). Recycle is also preferred over incineration since it helps reducing the battery's environmental impact (Yu et al. 2014).

Regarding greenhouse gases (GHG) emitted as a result of recharging a battery, they strongly depend on the energy source employed (Hawkins, Gausen, and Stromman 2012). The analysis of a battery aging is bound to help determining the moment at which it has to be recycled, thus, reduce its environmental impact, as well as improving its life.

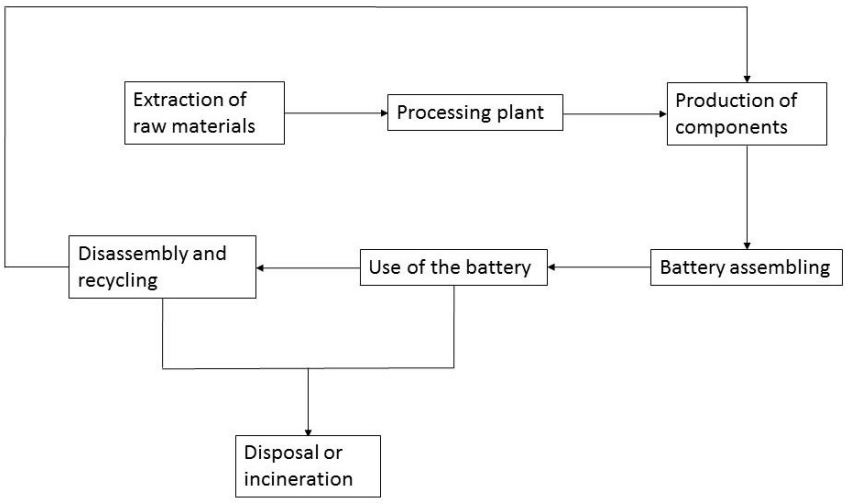

Figure 51. Battery life cycle, from (Van den Bossche et al. 2006).

Chapter 5 Conclusions and future work

This chapter presents conclusions of this project and provides hints for future work. Overall, the CFD approach makes the numerical simulation faster and further dynamic simulation could be carried out, such as: hybrid vehicle simulations, battery design, battery cycles, management and power control (W. Gu, Wang, and Liaw 1997).

5.1. Conclusions

A mathematical model of a Ni-MH battery was developed in Matlab®, such model may become a powerful tool to assess battery performance in future projects. The simulations prove that operating conditions can be modified and tested, i.e., complex coupled differential equations are effectively solved. Parameters such as concentration and potential profiles can be solved using a single solver within the numerical framework.

Due to the lack of the inclusion of side reactions which are important during charging (W. B. Gu and Wang 2000a; Gu, W. B. Wang 1998), the analysis has been restricted to the discharge performance of the cell for both isothermal and non-isothermal scenarios. Nonetheless, modelling side reactions was not a target of this project. All these capabilities are fundamental for battery research to resolve and understand not only the behaviour itself, but also performance limiting situations or failure mechanisms.

The model is in good agreement with literature and therefore validated. Also, the inclusion of the temperature variation is given as valid, though comparisons with previous developed models cannot be set, it has been found that under different conditions the performance is as expected.

Also, charge and discharge cycles have been included in order to conduct the entropy analysis. Entropy has been, then used as a degradation indicator by assessing the irreversible entropy, i.e., the entropy generated during operation that is lost. Simulations were run varying several parameters to analyse its performance, determining that the major contributions are due to the decrease of porosity in the electrodes. However, when studying the variation of entropy during discharge/charge cycles, the decay of the maximum concentration of the intercalating species contribution is larger than the deterioration of porosity.

5.2. Future work

Despite the decent agreement with other models, further modelling is required to improve the model.

Firstly, the solid phase equation should be computed in each point of the mesh. Instead of using valid approximations (DeVidts, Delgado, and White 1995; Gu, W. B. Wang 1998). In that way, side reactions should be taken into account to further study degradation (Albertus, Christensen, and Newman 2008).

This would also enable further thermal analysis (W. B. Gu and Wang 2000a). Furthermore, The lumped model approach presented in Chapter 1, should be modelled and compared with the per basis thermal model in order to explore the agreement between both models.

Regarding the study of degradation, a transient equation following the general form showed in Chapter 2 could be employed to model porosity decay. This would need of experimental data regarding porosity's degradation.

Also, a 2D and 3D mathematical and thermal model should be carried out to fully account for the changes inside the battery. Furthermore, it would be riveting an accuracy vs computational time comparison to determine whether it is worth to account for a 2D or 3D model.

This work, could be a starting point to model Li-ion batteries since the experimental data available in the department is of this type. It is also true that li-ion battery modelling is far more complex than metal hydride, notwithstanding (W. B. Gu and Wang 2000b)present a li-ion model similar to those presented for MH which should be attainable using the tools from this project.

In that way, a benchmark of experimental data should be formed to avoid corroboration against other author's models, this means: series of cell voltage for different C-rates, OCV and current. Likewise, experimental tests could be carried out to measure electrode's and electrolyte's properties.

Finally, a model for cells in a stack could be developed taking this work as a basis. A cell stack is a certain number of cells put together, with heat dissipation and the amount of cells used as primary concerns.

Economic Assessment

The cost of man hours as well as hardware and software equipment are presented in table 5. Nine month were dedicated on the project from April to December, 2016, three of them were on part time and the others full time yielding to a total of 1200 man hours. Fixing the price per work hour as 15€, the cost of engineering sums to 18000€. This includes all the simulations required for the study. Power consumption, i.e. the cost of having the computer operational, adds 60€, the electric power consumption price is estimated as 0,05€. Hardware and software amortizations correspond to the partial cost of licenced programmes and tools used in this project sum up to 72€. Guidance from advisors is accounted as consultancy, about ten meetings have been held throughout the project, it has been estimated 100€ per meeting regardless it was with one or both advisors, also e-mail consultancy has been included in the price. The total cost is 19024€.

Table 5. Budget of this project.

Concept	h	€/h	Cost (€)
Work hours			
Model developing	600	15	9000
Simulations	400	15	6000
Report	200	15	3000
Subtotal worked hours	1200		1200
Software and hardware			
Power consumption	1200	0,05	60
PC (i7-3630QM @2.4GHz)	1200	0,06	72
Matlab	700	0,06	42
Word	500	0,06	30
Subtotal software and hardware			*72*
Consultancy	10	100	1000
Total			**19204**

Bibliography

Albertus, Paul, John Christensen, and John Newman. 2008. “Modeling Side Reactions and Nonisothermal Effects in Nickel Metal-Hydride Batteries.” *Journal of The Electrochemical Society* 155 (1): A48. doi:10.1149/1.2801381.

Bard, A. J., and L. R. Faulkner. 2001. *ELECTROCHEMICAL METHODS: Fundamentals and Applications*. doi:10.1146/annurev.matsci.30.1.117.

Bernardi, D. 1985. “A General Energy Balance for Battery Systems.” *Journal of The Electrochemical Society* 132 (1): 5. doi:10.1149/1.2113792.

Bryant, Michael D. 2014. “Modeling Degradation Using Thermodynamic Entropy.” *Phm*, no. 1: 1–7.

Chehab, Z., L. Serrao, Y.G. Guezennec, and G. Rizzoni. 2006. “Aging Characterization of Nickel - Metal Hydride Batteries Using Electrochemical Impedance Spectroscopy.” *American Society of Mechanical Engineers, Dynamic Systems and Control Division (Publication) DSC*, 1–7. doi:10.1115/IMECE2006-14608.

COMSOL. 2011. “1D Isothermal Nickel Metal Hydride Battery.” https://www.comsol.com/model/1d-isothermal-nickel-metal-hydride-battery-9435.

Cuadras, Angel, Javier Cris??stomo, Victoria J. Ovejas, and Marcos Quilez. 2015. “Irreversible Entropy Model for Damage Diagnosis in Resistors.” *Journal of Applied Physics* 118 (16). doi:10.1063/1.4934740.

Cuadras, Angel, Victoria J. Ovejas, and Marcos Quilez. 2013. “Entropy as a Wear out Indicator: The Resistor Example.” *2013 10th International Multi-Conference on Systems, Signals and Devices, SSD 2013*, 1–5. doi:10.1109/SSD.2013.6564142.

DeVidts, P, J Delgado, and R E White. 1995. “Mathematical Modeling for the Discharge of a Metal Hydride Electrode.” *Journal of the Electrochemical Society* 142 (12): 4006–13. doi:10.1149/1.2048454.

Ebner, M., and V. Wood. 2015. “Tool for Tortuosity Estimation in Lithium Ion Battery Porous Electrodes.” *Journal of the Electrochemical Society* 162 (2): A3064–70. doi:10.1149/2.0111502jes.

European Commission DG Environment New Alert Service. 2012. “Environmental Impacts of Batteries for Low Carbon Technologies Compared.” *Science for Environment Policy*, 1. doi:10.1016.

Fuller, Thomas F. Doyle, Marc. Newman, John. 1994a. “Simulation and Optimization of the Dual Lithium Ion Insertion Cell.” *Journal of The Electrochemical Society* 141 (1): 1. doi:10.1149/1.2054684.

———. 1994b. “The Importance of the Lithium Ion Transference Number in Lithium/polymer Cells.”

Gu, W. B. Wang, C. 1998. “Micro-Macroscopic Coupled Modeling of Batteries and Fuel Cells.” *Journal of The Electrochemical Society* 145 (10): 3418. doi:10.1149/1.1838821.

Gu, W. B., and C. Y. Wang. 2000a. “Thermal-Electrochemical Modeling of Battery Systems.” *Journal of*

The Electrochemical Society 147 (8): 2910. doi:10.1149/1.1393625.

Gu, W B, and C Y Wang. 2000b. "Thermal and Electrochemical Coupled Modeling of a Lithium-Ion Cell, in Lithium Batteries." *ECS Proceedings* 99–25 (1): 748–62.

Gu, Wen-bin, Chao-yang Wang, and Bor Yann Liaw. 1997. "NUMERICAL SIMULATION OF COUPLED ELECTROCHEMICAL AND TRANSPORT PROCESSES IN BATTERY SYSTEMS." *Proceedings of the Intersociety Energy Conversion Engineering Conference* 2: 899–903.

Hawkins, Troy R., Ola Moa Gausen, and Anders Hammer Stromman. 2012. "Environmental Impacts of Hybrid and Electric Vehicles-a Review." *International Journal of Life Cycle Assessment* 17 (8): 997–1014. doi:10.1007/s11367-012-0440-9.

Hejabi, M., A. Oweisi, and N. Gharib. 2006. "Modeling of Kinetic Behavior of the Lead Dioxide Electrode in a Lead-Acid Battery by Means of Electrochemical Impedance Spectroscopy." *Journal of Power Sources* 158 (2 SPEC. ISS.): 944–48. doi:10.1016/j.jpowsour.2005.11.054.

Hu, Xiaosong, Jiuchun Jiang, Dongpu Cao, and Bo Egardt. 2016. "Battery Health Prognosis for Electric Vehicles Using Sample Entropy and Sparse Bayesian Predictive Modeling." *IEEE Transactions on Industrial Electronics* 63 (4): 2645–56. doi:10.1109/TIE.2015.2461523.

Kiehne, H A. 2003. *Battery Technology Handbook*. doi:10.1016/0378-7753(90)80038-F.

Kopera, John J C. 2004. "Inside the Nickel Metal Hydride Battery Inside the NiMH Battery." *Http://Www.Cobasys.Com/Pdf/Tutorial/Inside_Nimh_Battery_Technology.Pdf*, no. June: Last Visited: 08/06/15. doi:10.1016/S0378-7753(03)00355-0.

Miró Jané, Arnau. 2014. "Study of Advanced Materials for Thermal Insulation in the Inner Solar System." http://upcommons.upc.edu/pfc/handle/2099.1/24422.

Newman, J. 2015. *Electrochemical Systems*. *Aging*. Vol. 7. doi:10.1017/CBO9781107415324.004.

Newman, John. 1968. "Engineering Design of Electrochemical Systems." *Industrial & Engineering Chemistry* 60 (4): 12–27. doi:10.1021/ie50700a005.

Newman, John, Karen Thomas, Hooman Hafezi, and Dean Wheeler. 2003. "Modeling of Lithium-Ion Batteries." *Journal of Power Sources*.

Pals, Carolyn R., and John Newman. 1995. "Thermal Modeling of the Lithium / Polymer Battery - II. Temperature Profiles in Cell Stack." *Journal of The Electrochemical Society* 142 (10): 3282–88. doi:10.1149/1.2049975.

Patankar, Suhas. 1980. *Numerical Heat Transfer and Fluid Flow*. *Series in Coputational Methods in Mechanics and Thermal Sciences*. doi:10.1016/j.watres.2009.11.010.

Paxton, B.K. 1995. "Mathematical Modeling of the Nickel/Metal Hydride Battery System."

Paxton, Blaine, and John Newman. 1997. "Modeling of Nickel / Metal Hydride Batteries." *Journal of The Electrochemical Society* 144 (11): 3818–31. doi:10.1149/1.1838098.

Rahn, Cd Christopher D., and Chao-Yang Cy Wang. 2012. *Battery Systems Engineering*. doi:10.1002/9781118517048.

Rao, Lin. 1997. "Heat-Generation Rate and General Energy Balance for Insertion Battery Systems." *Journal of The Electrochemical Society* 144 (8): 2697. doi:10.1149/1.1837884.

Roache, Patrick J. 2002. "Code Verification by the Method of Manufactured Solutions." *Journal of Fluids Engineering* 124 (1): 4. doi:10.1115/1.1436090.

Salari, Kambiz, Knupp, Patrick. 2000. *Code Verification by the Method of Manufactured Solutions*.

Schulze, Gustav. 1974. *Metallphysik*. 2nded. Springer-Verlag.

Serrao, Lorenzo, Zakaria Chehab, Yann Guezennec, and Giorgio Rizzoni. 2005. "An Aging Model of Ni-MH Batteries for Hybrid Electric Vehicles." *2005 IEEE Vehicle Power and Propulsion Conference, VPPC* 2005: 78–85. doi:10.1109/VPPC.2005.1554536.

Van den Bossche, Peter, Frédéric Vergels, Joeri Van Mierlo, Julien Matheys, and Wout Van Autenboer. 2006. "SUBAT: An Assessment of Sustainable Battery Technology." *Journal of Power Sources* 162 (2 SPEC. ISS.): 913–19. doi:10.1016/j.jpowsour.2005.07.039.

Wakui, Yoshito, and Katsutoshi Inoue. 1998. "Hydrometallurgical Process for Recovery of Metal Values from Spent Nickel-Metal Hydride Secondary Batteries Hydrometallurgical Process for Recovery of Metal Values from Spent Nickel-Metal Hydride Secondary Batteries." *Hydrometallurgy*, no. March 2014. doi:10.1016/S0304-386X(98)00046-2.

Wu, M.S., Y.Y. Wang, and C.C. Wan. 1998. "Thermal Behaviour of Nickel/metal Hydride Batteries during Charge and Discharge." *Journal of Power Sources* 74 (2): 202–10. doi:10.1016/S0378-7753(98)00064-0.

Young, Kwo-hsiung, and Shigekazu Yasuoka. 2016. "Capacity Degradation Mechanisms in Nickel/Metal Hydride Batteries." *Batteries* 2 (1): 3. doi:10.3390/batteries2010003.

Yu, Yajuan, Bo Chen, Kai Huang, Xiang Wang, and Dong Wang. 2014. "Environmental Impact Assessment and End-of-Life Treatment Policy Analysis for Li-Ion Batteries and Ni-MH Batteries." *International Journal of Environmental Research and Public Health* 11 (3): 3185–98. doi:10.3390/ijerph110303185.

Zhang, Pingwei, Toshiro Yokoyama, Osamu Itabashi, Yoshito Wakui, Toshishige M. Suzuki, and Katsutoshi Inoue. 1998. "Hydrometallurgical Process for Recovery of Metal Values from Spent Nickel-Metal Hydride Secondary Batteries." *Hydrometallurgy* 50 (1): 61–75. doi:10.1016/S0304-386X(98)00046-2.

Appendices-A Matlab® Code

A.1 Main code

This is the Matlab® code for the discharge of a metal hydride battery. The first lines of the code are the input parameters used to calculate the time dependent outputs, such as: cell voltage, concentrations, temperature and overpotential.

```
clear;clc;close all;

%Constants
R = 8.314472 ;%SI
F = 96485.3329; %SI
T_out = 298.15; %Temperature outside the cell, set  to 25°C
T_ref = 298.15; %Referemce temperature, set to 25°C
M_h2o = 18; %g/mole
M_koh = 56.1056;


% Parameters

% NEGATIVE ELECTRODE
rs_mh = 1.5e-4;
eps_mh_l = 0.396;
eps_mh_s = 0.481;
a_mh = 3*eps_mh_s/rs_mh;
rho_mh = 7.49;
sigma_mh = 1000; %S/cm
eps_sep = 0.5;
eps_ni_l = 0.387;
eps_ni_s =  0.5;
alphaA_mh = 0.25;
alphaC_mh = 0.54;
Dh_mh = 2e-8; %cm2/s
Ch_mh_max =0.1025;
Ch_mh_ref = Ch_mh_max / 2; %mol/cm3
iref_mh = 0.785e-3; %A/cm2

% SEPARATOR AND ELECTROLYTE

rho_sep = 0.9;
toh = 0.22; %KOH transport number
Coh_ref = 6.91e-3;
Ch2o_ref = 0.05;

% POSITIVE ELECTRODE
rs_ni = 2.5e-4; % cm
rho_ni = 3.55;
sigma_ni = 28;
iref_ni = 0.104e-3;
alphaA_ni = 0.13;
alphaC_ni = 0.074;
```

```
Ch_ni_max = 0.0383;
Ch_ni_ref = Ch_ni_max / 2;
a_ni = 3*eps_ni_s/rs_ni; % cm2/cm3

% BATTERY PARAMETERS
Q = 43.4e-3; %Ah/cm2
Crate =Q;
[SOC,E_neg] = importfile('eeq neg.txt',2, 100);
[SOC,E_pos] = importfile('eeq pos.txt',2, 100);


% Initial values
I_cell = Crate; %C*1
Ch_i_mh = 0.95*Ch_mh_max;
Ch_i_ni =0.01*Ch_ni_max;
Coh_i = 6.91e-3;
phiemh_i =0;
phiesep_i = 0;
phieni_i = 0;
phismh_i = 0;
phissep_i = 0;
phisni_i = 0;
T_i = 25+273.15; % Initial temperature
Tl = 298.15; % Lumped model initial temperature

% Thermal parameters
Ea_k = 13e3; %J/mol
Ea_sigma = 0;
Ea_doh = 14e3;
Ea_dh = 9.66e3;
Ea_imh = 30e3;
Ea_isep = 0;
Ea_ini = 20e3;
dudtmh = -1.125e-3; %mV/K
dudtni = -1.125e-3;
cpmh = 0.35;
cpsep = 1.9;
cpni = 0.88;
cpkoh = 3.2;
ht = 5e-4; %W/cm2K

% SOLVER PARAMETERS
epsilon = 1e-6;
maxiter = 9000;
krelax = 0.5; % relaxation factor (between [1,0])

% Mesh  parameters
n = 100; % Number of nodes
L_mh = 0.035;
L_sep = 0.025; % cm
L_ni = 0.0843;



% Time Parameters
dt = 10; %Time interval for simulations
t_enddisch = (1*Q*3600)/abs(I_cell); %seconds "for full discharge"
tlast =t_enddisch;
```

```
t = 0:dt:tlast; % Time vector for simulation


%Meshes, 2 Geometry 3 sets of properties
[Ea_i,iref,Ch_ref,Ch_max,Ch_se,rho,cp,Tmean,V_cell,nnod,x,xcv,Ax,Coh,eps_l,
eps_s,phis,phie,i_e,T,dudt,alphaA,alphaC,sigma,a_cell,Ra_mh,nnody_mh,y_mh,y
cv_mh,Ay_mh,Ch_mh,Dh_mh1,Ra_ni,nnody_ni,y_ni,ycv_ni,Ay_ni,Ch_ni] =
MESH1D(Ea_imh,Ea_isep,Ea_ini,Ch_ni_ref,Ch_mh_ref,iref_mh,iref_ni,Ch_ni_max,
Ch_mh_max,rs_mh,rs_ni,T_i,L_mh,L_sep,L_ni,a_mh,a_ni,sigma_mh,sigma_ni,alpha
A_mh,alphaA_ni,alphaC_mh,alphaC_ni,Coh_i,phismh_i,phissep_i,phisni_i,phiemh
_i,phiesep_i,phieni_i,dudtmh,n,dudtni,eps_mh_l,eps_ni_l,eps_mh_s,
eps_sep,eps_ni_s,cpmh, cpsep,
cpni,rho_mh,rho_sep,rho_ni,Ch_i_mh,Ch_i_ni,Dh_mh);


% CALL BATTERY MODEL

[DOD,Tmean,V_cell,Utilization,Vloss_mh,Vloss_sep,Vloss_ni,T,Coh,Coh_M,Ch_se
,Ch_adim,Ch_mh,Ch_ni,phis,phie,i_e,Biot,ScT]=ONEDMHBATTERY(cpkoh,eps_ni_s,e
ps_ni_l,eps_sep,eps_mh_s,eps_mh_l,dt,rs_mh,rs_ni,krelax,Coh_ref,Ch2o_ref,T_
ref,T_out,Ch_mh_max,Ch_ni_max,R,F,Ea_k,Ea_sigma,Ea_doh,Ea_dh,Ea_i,M_h2o,M_k
oh,toh,SOC,E_neg,E_pos,maxiter,Q,I_cell,t,epsilon,ht,iref,Ch_ref,Ch_max,Ch_
se,rho,cp,Tmean,V_cell,nnod,x,Ax,Coh,eps_l,eps_s,phis,phie,i_e,T,dudt,alpha
A,alphaC,sigma,a_cell,Ra_mh,nnody_mh,y_mh,Ay_mh,Ch_mh,Dh_mh1,Ra_ni,nnody_ni
,y_ni,Ay_ni,Ch_ni,...
    'discharge','isothermal');
```

A.2 Battery computational toolbox

These are the different modules created by the author that comprise the toolbox used in this project.

```
function
[Ea_i,iref,Ch_ref,Ch_max,Ch_se,rho,cp,Tmean,V_cell,nnod,x,xcv,Ax,Coh,eps_l,
eps_s,phis,phie,i_e,T,dudt,alphaA,alphaC,sigma,a_cell,Ra_mh,nnody_mh,y_mh,y
cv_mh,Ay_mh,Ch_mh,Dh_mh1,Ra_ni,nnody_ni,y_ni,ycv_ni,Ay_ni,Ch_ni] =
MESH1D(Ea_imh,Ea_isep,Ea_ini,Ch_ni_ref,Ch_mh_ref,iref_mh,iref_ni,Ch_ni_max,
Ch_mh_max,rs_mh,rs_ni,T_i,L_mh,L_sep,L_ni,a_mh,a_ni,sigma_mh,sigma_ni,alpha
A_mh,alphaA_ni,alphaC_mh,alphaC_ni,Coh_i,phismh_i,phissep_i,phisni_i,phiemh
_i,phiesep_i,phieni_i,dudtmh,n,dudtni,eps_mh_l,eps_ni_l,eps_mh_s,
eps_sep,eps_ni_s,cpmh, cpsep,
cpni,rho_mh,rho_sep,rho_ni,Ch_i_mh,Ch_i_ni,Dh_mh)
%%MESH1D Puts listed parameters into suited form to create a 1D mesh.
%
% This function uses MeshNC1D property of:
%   Arnau Miró, UPC-ETSEIAT 2013-2014
%  and MeshNC1DY, a modified MeshNC1D to suit for spherical coordinates.
%
% Input:
% The inputs are the non-variable/initial values of the cell.
% Should more parameters be needed, they ought to be added manually.
% To avoid mistakes, recall that inputs need to be into slab form
```

```
% [neg,sep,pos]
%
%   Output:
% The outputs are the parameters returned into mesh form.
%
%   This code is part of the COMPUTATIONAL BATTERY TOOLBOX
%   Pol Miró, UPC-EEBE 2016-2017


% Preallocating
T = [T_i,T_i,T_i];
L_cell = [L_mh,L_sep,L_ni];
a_cell = [a_mh,0,a_ni];
sigma = [sigma_mh,0,sigma_ni];
alphaA = [alphaA_mh,0,alphaA_ni];
alphaC = [alphaC_mh,0,alphaC_ni];
Coh =[Coh_i,Coh_i,Coh_i];
phis =[phismh_i,phissep_i,phisni_i];
phie =[phiemh_i,phiesep_i,phieni_i];
i_e_i =[0,0,0];
dudt = [dudtmh,0,dudtni];
ncell = [n,n,n];
eps_l = [eps_mh_l, eps_mh_l,eps_ni_l];
eps_s = [eps_mh_s, eps_sep,eps_ni_s];
cpcell = [cpmh, cpsep, cpni];
rhocell =[rho_mh,rho_sep,rho_ni];
Chcell = [Ch_i_mh,0,Ch_i_ni];
Chcellmax = [Ch_mh_max,0,Ch_ni_max];
Chcellref =[Ch_mh_ref,0,Ch_ni_ref];
irefcell =[iref_mh,0,iref_ni];
Ea_cell =[Ea_imh,Ea_isep,Ea_ini];

Tmean = 0;
V_cell = 0;


%Meshes, 2 Geometry 3 sets of properties
[nnod,x,xcv,Ax,Coh,eps_l,eps_s,phis,phie,i_e,T,dudt,alphaA,alphaC,sigma,a_c
ell,cp,rho,Ch_se,Ch_max,Ch_ref,iref,Ea_i] =
MeshNC1D(L_cell,ncell,Coh,eps_l,eps_s,phis,phie,i_e_i,T,dudt,alphaA,alphaC,
sigma,a_cell,cpcell,rhocell,Chcell,Chcellmax,Chcellref,irefcell,Ea_cell);
%Global mesh
[Ra_mh,nnody_mh,y_mh,ycv_mh,Ay_mh,Ch_mh,Dh_mh1] = MeshNC1DY(rs_mh,n-
1,Ch_i_mh,Dh_mh);
[Ra_ni,nnody_ni,y_ni,ycv_ni,Ay_ni,Ch_ni] = MeshNC1DY(rs_ni,n-1,Ch_i_ni);

end


function
[DOD,Tmean,V_cell,Utilization,Vloss_mh,Vloss_sep,Vloss_ni,T,Coh,Coh_M,Ch_se
,Ch_adim,Ch_mh,Ch_ni,phis,phie,i_e,Biot,ScT]=ONEDMHBATTERY(cpkoh,eps_ni_s,e
ps_ni_l,eps_sep,eps_mh_s,eps_mh_l,dt,rs_mh,rs_ni,krelax,Coh_ref,Ch2o_ref,T_
ref,T_out,Ch_mh_max,Ch_ni_max,R,F,Ea_k,Ea_sigma,Ea_doh,Ea_dh,Ea_i,M_h2o,M_k
oh,toh,SOC,E_neg,E_pos,maxiter,Q,I_cell,t,epsilon,ht,iref,Ch_ref,Ch_max,Ch_
se,rho,cp,Tmean,V_cell,nnod,x,Ax,Coh,eps_l,eps_s,phis,phie,i_e,T,dudt,alpha
```

```
A,alphaC,sigma,a_cell,Ra_mh,nnody_mh,y_mh,Ay_mh,Ch_mh,Dh_mh1,Ra_ni,nnody_ni
,y_ni,Ay_ni,Ch_ni,varargin)
%ONEDMHBATTERY solve 1D battery during discharge.
%
%  It can be chosen whether the discharge is made under isothermal or non-
isothermal conditions
%
%   Output:
%   It returns the main characteristics of a battery.
%
%   This code is part of the COMPUTATIONAL BATTERY TOOLBOX
%   Pol Miró, UPC-EEBE 2016-2017

ind = find(strcmp(varargin,'discharge'));
switch lower(varargin{ind+1})
    case 'nonisothermal'
% Figure plot per timestep
fig = figure('Name','Plot per timestep','Color','w');
ax(1) = subplot(2,2,1,'Nextplot','Add'); grid on; axis ([0 x(nnod) 0 1]);
ylabel('Ch_{MH}');
ax(2) = subplot(2,2,2,'Nextplot','Add'); grid on; axis ([0 x(nnod) 6 8]);
ylabel('Coh');
ax(3) = subplot(2,2,3,'Nextplot','Add'); grid on; axis ([0 1 0.85 1.5]);
ylabel('V_cell');
ax(4) = subplot(2,2,4,'Nextplot','Add'); grid on; axis ([0 1 20 80]);
ylabel('T');
for i = 1:length(t) %Time loop

% Initializing
Tlast = T;
Cohlast = Coh;
Ch_mh_last = Ch_mh;
Ch_ni_last = Ch_ni;
phislast = phis;
phielast = phie;
DOD(i,1) = (abs(I_cell)*t(i))/(Q*3600);

for ii = 1:maxiter %3rd Loop
 % Update parameter values
 [kd,N_oh1,Ue3,Doh_eff,Dh_mh,Dh_ni,U,rho_koh,sigma_eff,Ch2o,k_eff] =
LIP(dudt,x,Dh_mh1,SOC,E_neg,E_pos,nnod,M_koh,M_h2o,y_ni,sigma,R,T,F,toh,Coh
,eps_l,eps_s,Ch_se,Ch_ni,Ch_mh_max,Ch_ni_max,T_ref,Ea_k,Ea_sigma,Ea_doh,Ea_
dh);
 [eta,i0,i_e] =
Currentdistribution(nnod,a_cell,Ea_i,F,R,T_ref,T,phis,phie,U,Coh,Coh_ref,al
phaC,alphaA,Ch_se,Ch_ref,Ch2o,Ch2o_ref,Ch_max,iref);

  for jj=1:maxiter  %2nd loop
        for j =1:maxiter %1st loop
 % Solve potential equations

[deltaphie,deltaphis,phis,phie]=PCS(phielast,k_eff,Ue3,krelax,x,i_e,phis,a_
cell,i0,F,alphaA,U,phie,R,T,alphaC,nnod,Ax,sigma_eff,phislast,I_cell);

 % CORRECT I_e and ETA
 i_elast = i_e;
```

```
 [eta,i0,i_e] =
Currentdistribution(nnod,a_cell,Ea_i,F,R,T_ref,T,phis,phie,U,Coh,Coh_ref,al
phaC,alphaA,Ch_se,Ch_ref,Ch2o,Ch2o_ref,Ch_max,iref);
 deltai_e = max(abs(i_e-i_elast));
if( deltai_e < epsilon),break;end % STOP CONDITION
        end
% SOLVE CONCENTRATIONS
[deltaChni,deltaChmh,deltaCoh,Coh,Coh_M,Ch_se,Ch_ni,Ch_mh,Ch_adim]=CCS(Ch_m
ax,a_cell,krelax,rs_mh,rs_ni,y_ni,y_mh,x,toh,F,i_e,nnod,Ax,dt,eps_l,Doh_eff
,Coh,Cohlast,nnody_ni,Ra_ni,Ay_ni,Dh_ni,Ch_ni,Ch_ni_last,nnody_mh,Ra_mh,Ay_
mh,Dh_mh,Ch_mh,Ch_mh_last);
if (deltaChni < epsilon && deltaCoh < epsilon && deltaChmh < epsilon),
break; end %STOP CONDITION
  end
for jjj =1:5 %4th loop
%THERMAL BALANCE
[deltaT,T,Biot,ScT]=HeatTransfer(I_cell,U,kd,krelax,eps_ni_s,eps_mh_l,eps_n
i_l,eps_sep,eps_mh_s,Ax,dt,Tlast,ht,T_out,phis,phie,x,nnod,rho,cp,eps_s,eps
_l,rho_koh,cpkoh,i_e,eta,T,dudt,sigma_eff,k_eff,N_oh1,...
                                    'mode','discretised');
if(deltaT < epsilon),break;end %STOP CONDITION
end

if (deltaT < epsilon && deltaChni < epsilon && deltaCoh < epsilon &&
deltaChmh < epsilon), break; end  % Exit condition

end
 % KEY PARAMETERS
Tmean(i,1) = ((T(1)+T(nnod))/2)-273.15;
V_cell(i,1) = phis(nnod) - phis(1);
Utilization(i,1) = mean(Ch_se(round(2*nnod/3)+1:nnod))/Ch_ni_max;
Vloss_mh(i,1) = -phis(1) +phie(round(nnod/3)) - 0.9722;
Vloss_sep(i,1) = phie(round(nnod/3)+1)-0.9722;
Vloss_ni(i,1) = phis(nnod)-phie(round(2*nnod/3)+1)-0.3878;

Sirr1(i,1) = (V_cell*I_cell)/Tmean;
Sirr2(i,1) = (trapz(eta,x)*I_cell)/Tmean;

 % Showing results


    % Plots per timestep
    if exist('plt','var'), delete(plt); end
    plt(1) = plot(ax(1),x,Ch_adim,'k');
    plt(2) = plot(ax(2),x,Coh_M,'k');
    plt(3) = plot(ax(3),DOD,V_cell,'k');
    plt(4) = plot(ax(4),DOD,Tmean,'k');
    drawnow;

if (Tmean(i)>80);break;end %STOP CONDITION
if (V_cell(i) < 0.85);break;end %STOP CONDITION
end

    case 'isothermal'
% Figure plot per timestep
fig = figure('Name','Plot per timestep','Color','w');
ax(1) = subplot(2,2,1,'Nextplot','Add'); grid on; axis ([0 x(nnod) 0 1]);
ylabel('Ch_{MH}');
```

```
ax(2) = subplot(2,2,2,'Nextplot','Add'); grid on; axis ([0 x(nnod) 6 8]);
ylabel('Coh');
ax(3) = subplot(2,2,3,'Nextplot','Add'); grid on; axis ([0 1 0.85 1.5]);
ylabel('V_cell');
ax(4) = subplot(2,2,4,'Nextplot','Add'); grid on; axis ([0 1 20 80]);
ylabel('T');
for i = 1:length(t) %Time loop

% Initializing
Tlast = T;
Cohlast = Coh;
Ch_mh_last = Ch_mh;
Ch_ni_last = Ch_ni;
phislast = phis;
phielast = phie;
DOD(i,1) = (abs(I_cell)*t(i))/(Q*3600);

for ii = 1:maxiter %3rd Loop
 % Update parameter values
 [kd,N_oh1,Ue3,Doh_eff,Dh_mh,Dh_ni,U,rho_koh,sigma_eff,Ch2o,k_eff] =
LIP(dudt,x,Dh_mh1,SOC,E_neg,E_pos,nnod,M_koh,M_h2o,y_ni,sigma,R,T,F,toh,Coh
,eps_l,eps_s,Ch_se,Ch_ni,Ch_mh_max,Ch_ni_max,T_ref,Ea_k,Ea_sigma,Ea_doh,Ea_
dh);
 [eta,i0,i_e] =
Currentdistribution(nnod,a_cell,Ea_i,F,R,T_ref,T,phis,phie,U,Coh,Coh_ref,al
phaC,alphaA,Ch_se,Ch_ref,Ch2o,Ch2o_ref,Ch_max,iref);

  for jj=1:maxiter  %2nd loop
        for j =1:maxiter %1st loop
 % Solve potential equations

[deltaphie,deltaphis,phis,phie]=PCS(phielast,k_eff,Ue3,krelax,x,i_e,phis,a_
cell,i0,F,alphaA,U,phie,R,T,alphaC,nnod,Ax,sigma_eff,phislast,I_cell);

 % CORRECT I_e and ETA
 i_elast = i_e;
 [eta,i0,i_e] =
Currentdistribution(nnod,a_cell,Ea_i,F,R,T_ref,T,phis,phie,U,Coh,Coh_ref,al
phaC,alphaA,Ch_se,Ch_ref,Ch2o,Ch2o_ref,Ch_max,iref);
 deltai_e = max(abs(i_e-i_elast));
if( deltai_e < epsilon),break;end % STOP CONDITION
        end
% SOLVE CONCENTRATIONS
[deltaChni,deltaChmh,deltaCoh,Coh,Coh_M,Ch_se,Ch_ni,Ch_mh,Ch_adim]=CCS(Ch_m
ax,a_cell,krelax,rs_mh,rs_ni,y_ni,y_mh,x,toh,F,i_e,nnod,Ax,dt,eps_l,Doh_eff
,Coh,Cohlast,nnody_ni,Ra_ni,Ay_ni,Dh_ni,Ch_ni,Ch_ni_last,nnody_mh,Ra_mh,Ay_
mh,Dh_mh,Ch_mh,Ch_mh_last);
if (deltaChni < epsilon && deltaCoh < epsilon && deltaChmh < epsilon),
break; end %STOP CONDITION
  end
for jjj =1:5 %4th loop
%THERMAL BALANCE
[deltaT,T,Biot,ScT]=HeatTransfer(I_cell,U,kd,krelax,eps_ni_s,eps_mh_l,eps_n
i_l,eps_sep,eps_mh_s,Ax,dt,Tlast,ht,T_out,phis,phie,x,nnod,rho,cp,eps_s,eps
_l,rho_koh,cpkoh,i_e,eta,T,dudt,sigma_eff,k_eff,N_oh1,...
                                    'mode','discretised');
T(:)=T_ref;
Biot = 0;
```

```
end

if (deltaChni < epsilon && deltaCoh < epsilon && deltaChmh < epsilon),
break; end  % Exit condition

end
 % KEY PARAMETERS
Tmean(i,1) = ((T(1)+T(nnod))/2)-273.15;
V_cell(i,1) = phis(nnod) - phis(1);
Utilization(i,1) = mean(Ch_se(round(2*nnod/3)+1:nnod))/Ch_ni_max;
Vloss_mh(i,1) = -phis(1) +phie(round(nnod/3)) - 0.9722;
Vloss_sep(i,1) = phie(round(nnod/3)+1)-0.9722;
Vloss_ni(i,1) = phis(nnod)-phie(round(2*nnod/3)+1)-0.3878;

 % Showing results


    % Plots per timestep
    if exist('plt','var'), delete(plt); end
    plt(1) = plot(ax(1),x,Ch_adim,'k');
    plt(2) = plot(ax(2),x,Coh_M,'k');
    plt(3) = plot(ax(3),DOD,V_cell,'k');
    plt(4) = plot(ax(4),DOD,Tmean,'k');
    drawnow;

if (Tmean(i)>80);break;end %STOP CONDITION
if (V_cell(i) < 0.85);break;end %STOP CONDITION
end
otherwise
error('Boundary %s not recognized',varargin{ind+1});
end
end

function [kd,N_oh1,Ue3,Doh_eff,Dh_mh,Dh_ni,U,rho_koh,sigma_eff,Ch2o,k_eff]
=
LIP(dudt,x,Dh_mh1,SOC,E_neg,E_pos,nnod,M_koh,M_h2o,y_ni,sigma,R,T,F,toh,Coh
,eps_l,eps_s,Ch_se,Ch_ni,Ch_mh_max,Ch_ni_max,T_ref,Ea_k,Ea_sigma,Ea_doh,Ea_
dh)

%%LIP calculates internal loop parameters.
%
%   Output:
% The function returns effective variables (BRUGGEMAN relation and
% TEMPERATURE dependence) and dummy variables needed elsewhere.
%
% Please note that if any concentration variable is desired to be taken as
% fixed value, then a (%) must be placed before the 'old' value and replace
% it by the new value in a proper format.
%
%   This code is part of the COMPUTATIONAL BATTERY TOOLBOX
%   Pol Miró, UPC-EEBE 2016-2017


% Inside loop parameters re-/calculation
    Doh = 2.8509e-5 -(2.9659e-4).*Coh.^(1./2)+0.013768.*Coh-
0.14199.*Coh.^(3./2)+0.42661.*Coh.^2;
    Doh_eff1 = Doh.*eps_l.^0.5;
```

```
    Dh_ni1 = 3.4e-8.*(1-
0.95661.*(Ch_ni(end)./Ch_ni_max)).*ones(size(y_ni));
    rho_koh = 1.001 + 47.52.*Coh - 776.22.*Coh.^2;
    Ch2o = (rho_koh - Coh.*M_koh)/M_h2o;
    k_koh = 0.02325+210.95.*Coh-22077.*Coh.^2 +(6.2907e5).*Coh.^3;
    k_eff1 = k_koh.*eps_l.^1.5;
    f_act = 1.0004-36.23.*(Coh.^0.5)+1374.3.*Coh-
17850.7.*(Coh.^(3./2))+55406.*(Coh.^2)+(7.16856e5).*(Coh.^(5/2));
    lncoh = log(f_act.*Coh);
    sigma_eff1 = sigma.*eps_s;
    Umh1 =
interp1(SOC,E_neg,(Ch_se(1:round(nnod/3))./Ch_mh_max),'linear','extrap');
    Uni1=interp1(SOC,E_pos,(1-
(Ch_se(round(2*nnod/3)+1:nnod)./Ch_ni_max)),'linear','extrap');

% Compute temperature dependences
   k_eff = k_eff1.*exp(((Ea_k./R).*((1./T_ref)-(1./T))));
   sigma_eff = sigma_eff1.*exp(((Ea_sigma./R).*((1./T_ref)-(1./T))));
   Doh_eff = Doh_eff1.*exp(((Ea_doh./R).*((1./T_ref)-(1./T))));
   Dh_mh = Dh_mh1.*exp(((Ea_dh./R).*((1./T_ref)-
(1./mean(T(1:round(nnod/3)))))));
   Dh_ni=Dh_ni1.*exp(((Ea_dh./R).*((1./T_ref)-
(1./mean(T(round(2*nnod/3)+1:nnod))))));
   a = 0.*x(1:((round(2*nnod/3)-round(nnod/3))));
   U1 = [Umh1;a;Uni1];
   U = U1+(T-T_ref).*dudt;

% Computing gradients
    kd = ((k_eff.*R.*T)./F).*((toh)+Coh./Ch2o);
    N_kd = mygrad(kd,x);
    N_oh1= mygrad(lncoh,x);
    N_oh = mygrad(mygrad(lncoh,x),x);
    Ue3 = N_kd.*N_oh1 + N_oh.*kd;
end
```

```
function [eta,i0,i_e] =
Currentdistribution(nnod,a_cell,Ea_i,F,R,T_ref,T,phis,phie,U,Coh,Coh_ref,al
phaC,alphaA,Ch_se,Ch_ref,Ch2o,Ch2o_ref,Ch_max,iref)
%%Currentdistribution calculates the current distribution alongside the
cell, named as j[A/cm3].
%
%
%   This code is part of the COMPUTATIONAL BATTERY TOOLBOX
%   Pol Miró, UPC-EEBE 2016-2017


eta = phis-phie-U;
i0t=((((Coh)./(Coh_ref)).^alphaC).*((Ch_se./Ch_ref).^alphaC).*(((Ch2o)./(Ch
2o_ref)).^alphaA).*(((Ch_max-Ch_se)./(Ch_max-Ch_ref)).^alphaA)).*iref;
i0t((round(nnod/3)+1:round(2*nnod/3)))=0;
i0= i0t.*exp(((Ea_i./R).*((1./T_ref)-(1./T))));
i_e =a_cell.*i0.*(exp(((alphaA.*F.*(eta))./(R.*T)))-exp(((-
alphaC.*F.*(eta))./(R.*T))));
end

function
[deltaphie,deltaphis,phis,phie]=PCS(phielast,k_eff,Ue3,krelax,x,i_e,phis,a_
cell,i0,F,alphaA,U,phie,R,T,alphaC,nnod,Ax,sigma_eff,phislast,I_cell)
%%PCS Solve 1D coupled electrolyte and solid potential.
%
%
%
%   Output:
% This function returns the solid-phase potential and the electrolyte
potential
% as well as the error of each.
%
%   This code is part of the COMPUTATIONAL BATTERY TOOLBOX
%   Pol Miró, UPC-EEBE 2016-2017



% Computing Source Terms
Spphis =-(a_cell.*i0.*(F.*alphaA.*exp(-(F.*alphaA.*(U+phie -
phis))./(R.*T)) + F.*alphaC.*exp((F.*alphaC.*(U+phie -
phis))./(R.*T)))).*(R.*T);
Scphis = - i_e -Spphis.*phis;

% Solver call
a = ones(size(x));
   [phisnew,iterphis,errphis] =
Diffusion1D(nnod,Ax,1e99,a,sigma_eff,phis,phislast,Scphis,Spphis,...
                          'LeftBound','fixed',0,...
                          'RightBound','flux',-I_cell);
    deltaphis = max( abs(phisnew -phis) );
    phis = phis*(1-krelax) + krelax*phisnew; % Actualize value

% Computing Source Terms
Spphie =-(a_cell.*i0.*(F.*alphaA.*exp(-(F.*alphaA.*(U+phie -
phis))./(R.*T)) + F.*alphaC.*exp((F.*alphaC.*(U+phie -
phis))./(R.*T)))).*(R.*T);
Scphie = i_e +Ue3-Spphie.*phie;
```

```
% Solver call
    [phienew,iterphie,errphie] =
Diffusion1D(nnod,Ax,1e99,a,k_eff,phie,phielast,Scphie,Spphie,...
                          'LeftBound','flux',0,...
                          'RightBound','flux',0);
    deltaphie = max( abs(phienew -phie) );
    phie = phie*(1-krelax) + krelax*phienew; % Actualize value

end


function
[deltaChni,deltaChmh,deltaCoh,Coh,Coh_M,Ch_se,Ch_ni,Ch_mh,Ch_adim]=CCS(Ch_m
ax,a_cell,krelax,rs_mh,rs_ni,y_ni,y_mh,x,toh,F,i_e,nnod,Ax,dt,eps_l,Doh_eff
,Coh,Cohlast,nnody_ni,Ra_ni,Ay_ni,Dh_ni,Ch_ni,Ch_ni_last,nnody_mh,Ra_mh,Ay_
mh,Dh_mh,Ch_mh,Ch_mh_last)
%%CCS Solve 1D coupled concentration equations for the electrolyte and
%solid-phase.
%
%
%
%   Output:
% This function returns the solid-phase  and the electrolyte concentration
% as well as the error of each.
%
%   This code is part of the COMPUTATIONAL BATTERY TOOLBOX
%   Pol Miró, UPC-EEBE 2016-2017


% Compute Source terms
Spoh = zeros(size(x));
Scoh = (((-toh)./F).*i_e);

% Solver Call

    [Cohnew,iteroh,erroh] =
Diffusion1D(nnod,Ax,dt,eps_l,Doh_eff,Coh,Cohlast,Scoh,Spoh,...
                          'LeftBound','flux',0,...
                          'RightBound','flux',0);
    deltaCoh = max( abs(Cohnew -Coh) );
    Coh = Coh*(1-krelax) + krelax*Cohnew;
    Coh_M = Coh.*1000; % Molar concentration (M) of electrolyte

% Compute Source terms
Sph_ni = zeros(size(y_ni));
Sch_ni = zeros(size(y_ni));
Sph_mh = zeros(size(y_mh));
Sch_mh = zeros(size(y_mh));

% Define boundary conditions
bcoi = 0;
bcoend_ni =-mean(i_e((round(2*nnod/3)+1:nnod))) ./ (a_cell(nnod).*F);
bcoend_mh =-mean(i_e(1:round(nnod/3))) ./ (a_cell(1).*F);

% Solver call
a = ones(size(x));
```

```
    [Chni_new,iter_h_ni,err_h_ni] =
Diffusion1DY(nnody_ni,Ra_ni,Ay_ni,dt,a,Dh_ni,Ch_ni,Ch_ni_last,Sch_ni,Sph_ni
,...
                                'LeftBound','flux',bcoi,...
                                'RightBound','flux',bcoend_ni);
    deltaChni = max( abs(Chni_new - Ch_ni) );
    Ch_ni = Ch_ni*(1-krelax) + krelax*Chni_new; % Actualize value

    [Chmh_new,iter_h_mh,err_h_mh] =
Diffusion1DY(nnody_mh,Ra_mh,Ay_mh,dt,a,Dh_mh,Ch_mh,Ch_mh_last,Sch_mh,Sph_mh
,...
                                'LeftBound','flux',bcoi,...
                                'RightBound','flux',bcoend_mh);
    deltaChmh = max( abs(Chmh_new -Ch_mh) );
    Ch_mh = Ch_mh*(1-krelax) + krelax*Chmh_new; % Actualize value

% Average concentration alongside the cell

    Ch_mh_se = Ch_mh(nnody_mh) -
((rs_mh./5).*i_e(1:round(nnod/3))./(Dh_mh(1).*a_cell(1).*F);
    Ch_ni_se = Ch_ni(nnody_ni) -
((rs_ni./5).*i_e(round(2*nnod/3)+1:nnod))./(Dh_ni.*a_cell(nnod).*F);
    a = 0.*x(1:((round(2*nnod/3)-round(nnod/3))));
    Ch_se = [Ch_mh_se ;a;Ch_ni_se];
    Ch_adim = Ch_se./Ch_max;

end

function
[deltaT,T,Biot,ScT]=HeatTransfer(I_cell,U,kd,krelax,eps_ni_s,eps_mh_l,eps_n
i_l,eps_sep,eps_mh_s,Ax,dt,Tlast,ht,T_out,phis,phie,x,nnod,rho,cp,eps_s,eps
_l,rho_koh,cpkoh,i_e,eta,T,dudt,sigma_eff,k_eff,N_oh1,varargin)
%Heat Transfer Solve 1D heat transfer across the cell.
%
%  One can choose a discretised approach or the lumped model approach
(Newman,Rao 1997)
%
%   Output:
% This function returns the temperature profile across the cell, the biot
% number, the error and the generation rate in the form of heat source
% term.
%
%   This code is part of the COMPUTATIONAL BATTERY TOOLBOX
%   Pol Miró, UPC-EEBE 2016-2017

ind = find(strcmp(varargin,'mode')); % Find index inside varargin
switch lower(varargin{ind+1})
    case 'discretised' % Discretised model

dphis1 = mygrad(phis(1:round(nnod/3)),x(1:1:round(nnod/3)));
dphis2 =
mygrad(phis((round(nnod/3)+1):round(2*nnod/3)),x(((round(nnod/3)+1):round(2
*nnod/3))));
dphis3 = mygrad(phis(round(2*nnod/3)+1:nnod),x((round(2*nnod/3)+1:nnod)));
dphis = [dphis1;dphis2;dphis3];
dphie = mygrad(phie,x);
```

```
% Heat sources
Qirr = i_e.*eta;
Qrev = i_e.*T.*dudt;
Qs = sigma_eff.*dphis.^2;
Qe = k_eff.*dphie.^2 + kd.*N_oh1.*dphie;
Qpol = Qs+Qe;
SpT = zeros(size(x));
ScT = (Qirr+Qrev+Qpol);

% Boundaries and parameters
k_T_mh = (1.16.*(eps_mh_s)+0.57.*(eps_mh_l)).*ones(size(x)).*10e-2;
k_T_sep = (0.22.*(eps_sep)+0.57.*(eps_mh_l)).*ones(size(x)).*10e-2;
k_T_ni =  (1.14.*(eps_ni_s)+0.57.*(eps_ni_l)).*ones(size(x)).*10e-2;
k_T = [k_T_mh(1:round(nnod/3));k_T_sep(1:round(nnod/3)-
1);k_T_ni(1:round(nnod/3))];
rhocp = eps_s.*rho.*cp + rho_koh.*cpkoh.*eps_l;
bcoi =- (ht.*(T(1)-T_out));
bcoend = -(ht.*(T(nnod)-T_out));

% Solver Call
    [T_new,iterT,errT] =
Diffusion1D(nnod,Ax,dt,rhocp,k_T,T,Tlast,ScT,SpT,...
                            'LeftBound','flux',bcoi,...
                            'RightBound','flux',bcoend);
    deltaT = max( abs(T_new -T) );
    T = T*(1-krelax) + krelax*T_new;
    Biot = ht*(x(nnod)) ./ k_T;

    case 'lumped'
 Tl = mean(T);
 rhocp = eps_mh_s.*rho(1).*cp(1)+
eps_ni_s.*rho(nnod).*cp(nnod)+eps_sep.*rho(nnod/2).*cp(nnod/2)+
mean((rho_koh)).*cpkoh.*eps_mh_l+mean((rho_koh)).*cpkoh.*eps_ni_l;
 Qpol = (trapz(i_e(1:round(nnod/3)).*(U(1:round(nnod/3))-
T(1:round(nnod/3)).*dudt(1:round(nnod/3))),x(1:round(nnod/3)))+
trapz(i_e(round(2*nnod/3)+1:nnod).*(U(round(2*nnod/3)+1:nnod)-
T(round(2*nnod/3)+1:nnod).*dudt(round(2*nnod/3)+1:nnod)),x(round(2*nnod/3)+
1:nnod)));
 Qrev = (I_cell).*(phis(nnod)-phis(1));
 ScT = Qpol+Qrev + ((ht.*(mean(T)-T_out))/x(nnod));
 Biot = 0;
 %Balance
    T_new = (dt/mean(rhocp))*ScT + Tl;
    deltaT = (T_new-Tl)^2;
    Tl = Tl*(1-krelax) + krelax*T_new; % Actualize value
    T = Tl.*ones(size(x));

otherwise
error('Boundary %s not recognized',varargin{ind+1});
end

end


function [Co,iter,err] =
Diffusion1D(n,Ax,dt,eps,D,Co0,Colast,Sc,Sp,varargin)
%Diffusion 1D Solves an uni-dimensional equation in the form of dc/dt =
```

```
%d^2c/dx^2 +S. for one time instant (dt) using a finite differences
approach.
%   To recover steady state, set a large enough dt. For making the mesh
%   see NCMESH1D. The code features with an harmonic mean function of
%   thermal conductivity to compute various materials.
%
%   Modeling of equations has been done following the book of PATANKAR.
%
%   Input:
%       - n: Number of mesh nodes. Other inputs must be of length n.
%       - Ax: A n x 3 matrix with Ax, Axe, Axw the mesh spacing. Given by
%         the NCMESH1D routine.
%       - dt: Time step [s].
%       - eps: Porosity or equivalent property.
%       - D: Diffusion or equivalent property.
%       - C0: Starting guessed variable field.
%       - Clast: Previous instant variable field.
%       - Ar: Node aera [m] as a vector of length n.
%       - Sc: Internal heat sources  independent of the variable.
%       - Sp: Internal heat sources as a function of the variable.
%
%   Boundaries: boundaries are inputed by setting a keyword and then the
%   number of inputs the boundary requests. Keywords are 'LeftBound' for
%   the left boundary and 'RightBound' for the right boundary. They must be
%   followed by the boundary type. Accepted boundary types:
%       > Fixed: fixed temperature at boudary.
%               - T: Temperature [K] of the fixed boundary.
%       > Flux: fixed heat flux at boundary.
%       > Convection: fixed heat flux at boundary in the form of h(T-Tf).
%
%   Solver: solver can be specified by setting the flag 'Solver' then
%   followed by the function handle of the solver. Solvers for this
%   function expect the standard inputs of (A,b,x0,n), where A is the
%   system matrix. If no solver is specified, the standard matlab 'A\b'
%   command is used. As options for the solver the flag 'sparse' can be
%   used to indentify a sparse solver and then the precision can be set. A
%   general call for a solver is:
%       'Solver',@solver_handle,'sparse',precision
%
%   Output:
%       - C: Variable solved
%       - iter: Number of iterations the solver needed to achieve
%       convergence.
%       - err: Maximum error achieved.
%
%   This code uses a TDMA solver and a code discretization for heat
transfer
%   equation modified for this works needs, property of
%   Arnau Miro, UPC-ETSEIAT 2013-2014
%
%   This code is part of the COMPUTATIONAL BATTERY TOOLBOX
%   Pol Miró, UPC-EEBE 2016-2017

% Extract Ax, Axe and Axw
Axe = Ax(:,2); Axw = Ax(:,3); Ax = Ax(:,1);

% Preallocating
ap = zeros(n,1); ap0 = zeros(n,1);
```

```
ae = zeros(n,1); aw = zeros(n,1);
b = zeros(n,1);

% Inside Nodes discretization (Ref. PATANKAR cap 4.3-3)
ae(2:n-1) = (HarmonicMean(D(2:n-1),D(3:n),.5) ./ Axe(2:n-1));
aw(2:n-1) = (HarmonicMean(D(2:n-1),D(1:n-2),.5) ./ Axw(2:n-1));
ap0(2:n-1)   = (eps(2:n-1) .* Ax(2:n-1))./dt;
b(2:n-1)  = Sc(2:n-1).*Ax(2:n-1)+ ap0(2:n-1).*Colast(2:n-1);
ap(2:n-1) = ae(2:n-1) + aw(2:n-1) + ap0(2:n-1) - Sp(2:n-1).*Ax(2:n-1);

% BOUNDARIES
% Left boundary
ind = find(strcmp(varargin,'LeftBound')); % Find index inside varargin
switch lower(varargin{ind+1})
    case 'fixed' % Co = Co* at boundary
        % Find Tf
        C = varargin{ind+2};
        ae(1) = 0;
        aw(1) = 0;
        ap(1) = 1;
        b(1)  = C;

    case 'flux' % dC/dx = J* at boundary
        % Find J*
        J = varargin{ind+2};
        ae(1) = (HarmonicMean(D(1),D(2),.5)/Axe(1));
        aw(1) = 0;
        ap0(1)= (eps(1).* Ax(1))./dt;
        ap(1) = ae(1) + aw(1) + ap0(1) - Sp(1)*Ax(1);
        b(1)  = ap0(1)*Colast(1) + Sc(1)*Ax(1) + J;

    otherwise
        error('Boundary %s not recognized',varargin{ind+1});
end

% Right boundary
ind = find(strcmp(varargin,'RightBound')); % Find index inside varargin
switch lower(varargin{ind+1})
    case 'fixed' % C = C* at boundary
        % Find Tf
        C = varargin{ind+2};
        ae(n) = 0;
        aw(n) = 0;
        ap(n) = 1;
        b(n)  = C;

    case 'flux' % dC/dx = J* at boundary
        % Find J
        J = varargin{ind+2};
        ae(n) = 0;
        aw(n) = (HarmonicMean(D(n),D(n-1),.5)/Axw(n));
        ap0(n)= (eps(n).* Ax(n))./dt;
        ap(n) = ae(n) + aw(n) + ap0(n) - Sp(n)*Ax(n);
        b(n)  = ap0(n)*Colast(n) + Sc(n)*Ax(n) + J;

    otherwise
        error('Boundary %s not recognized',varargin{ind+1});
end
```

```
% Check for negative coefficients
if (any(ae)<0 || any(aw)<0 || any(ap)<0 || any(b)<0), keyboard; end

% Solver
Co = TDMA(ap,aw,ae,b,Co0,n);
iter = 1; err = 0;
end

function xi = HarmonicMean(xP,xI,f)

xi = ( (1-f)./xP + f./xI ).^(-1);

end


function [Co,iter,err] =
Diffusion1DY(n,Ra,Ax,dt,eps,D,Co0,Colast,Sc,Sp,varargin)
%Diffusion 1DY Solves an uni-dimensional equation in spherical coordinates
%for one time instant (dt) using a finite differences approach.
%   To recover steady state, set a large enough dt. For making the mesh
%   see NCMESH1D. The code features with an harmonic mean function of
%   thermal conductivity to compute various materials.
%
%   Modeling of equations has been done following the book of PATANKAR.
%
%   Input:
%       - n: Number of mesh nodes. Other inputs must be of length n.
%       - Ax: A n x 3 matrix with Ax, Axe, Axw the mesh spacing. Given by
%         the NCMESH1D routine.
%       - dt: Time step [s].
%       - eps: Porosity or equivalent property.
%       - D: Diffusion or equivalent property.
%       - C0: Starting guessed variable field.
%       - Clast: Previous instant variable field.
%       - Ar: Node aera [m] as a vector of length n.
%       - Sc: Internal heat sources  independent of the variable.
%       - Sp: Internal heat sources as a function of the variable.
%
%   Boundaries: boundaries are inputed by setting a keyword and then the
%   number of inputs the boundary requests. Keywords are 'LeftBound' for
%   the left boundary and 'RightBound' for the right boundary. They must be
%   followed by the boundary type. Accepted boundary types:
%       > Fixed: fixed temperature at boudary.
%               - T: Temperature [K] of the fixed boundary.
%       > Flux: fixed heat flux at boundary.
%       > Convection: fixed heat flux at boundary in the form of h(T-Tf).
%
%   Solver: solver can be specified by setting the flag 'Solver' then
%   followed by the function handle of the solver. Solvers for this
%   function expect the standard inputs of (A,b,x0,n), where A is the
%   system matrix. If no solver is specified, the standard matlab 'A\b'
%   command is used. As options for the solver the flag 'sparse' can be
%   used to indentify a sparse solver and then the precision can be set. A
%   general call for a solver is:
%       'Solver',@solver_handle,'sparse',precision
%
```

```
%   Output:
%       - C: Variable solved
%       - iter: Number of iterations the solver needed to achieve
%       convergence.
%       - err: Maximum error achieved.
%
%   This code uses a TDMA solver property
%   Arnau Miro, UPC-ETSEIAT 2013-2014
%
%   This code is part of the COMPUTATIONAL BATTERY TOOLBOX
%   Pol Miró, UPC-EEBE 2016-2017

% Extract Ax, Axe and Axw
Axe = Ax(:,2); Axw = Ax(:,3); Ax = Ax(:,1);

% Extract Rp,Re and Rw
Re = Ra(:,2); Rw = Ra(:,3) ; Rp = Ra(:,1);

% Preallocating
ap = zeros(n,1); ap0 = zeros(n,1);
ae = zeros(n,1); aw = zeros(n,1);
b = zeros(n,1);

% Inside Nodes discretization (Ref. PATANKAR cap 4.3-3)
ae(2:n-1) = (((Re(2:n-1)).^2).*(HarmonicMean(D(2:n-1),D(3:n),.5)) ./
(((Rp(2:n-1)).^2).*Axe(2:n-1)));
aw(2:n-1) = (((Rw(2:n-1)).^2).*(HarmonicMean(D(2:n-1),D(1:n-2),.5)) ./
(((Rp(2:n-1)).^2).*Axw(2:n-1)));
ap0(2:n-1)   =Ax(2:n-1)./dt;
b(2:n-1)  = Sc(2:n-1).*Ax(2:n-1)+ ap0(2:n-1).*Colast(2:n-1);
ap(2:n-1) = ae(2:n-1) + aw(2:n-1) + ap0(2:n-1) - Sp(2:n-1).*Ax(2:n-1);

% BOUNDARIES
% Left boundary
ind = find(strcmp(varargin,'LeftBound')); % Find index inside varargin
switch lower(varargin{ind+1})
    case 'fixed' % Co = Co* at boundary
        % Find Tf
        C = varargin{ind+2}; %Inicialment C(x,t)=Cinicial
        ae(1) = 0;
        aw(1) = 0;
        ap(1) = 1;
        b(1)  = C;

    case 'flux' % dC/dx = J* at boundary
        % Find J*
        J = varargin{ind+2};
        ae(1) =
(((Re(1)).^2)*(HarmonicMean(D(1),D(2),.5))/(((Rp(1)).^2).*Axe(1)));
        aw(1) = 0;
        ap0(1)= Ax(1)./dt;
        ap(1) = ae(1) + aw(1) + ap0(1) - Sp(1)*Ax(1);
        b(1)  = ap0(1)*Colast(1) + Sc(1)*Ax(1) + J;

    otherwise
        error('Boundary %s not recognized',varargin{ind+1});
end
```

```
% Right boundary
ind = find(strcmp(varargin,'RightBound')); % Find index inside varargin
switch lower(varargin{ind+1})
    case 'fixed' % C = C* at boundary
        % Find Tf
        C = varargin{ind+2};
        ae(n) = 0;
        aw(n) = 0;
        ap(n) = 1;
        b(n)  = C;

    case 'flux' % dC/dx = J* at boundary
        % Find J
        J = varargin{ind+2};
        ae(n) = 0;
        aw(n) = (((Rw(n)).^2).*(HarmonicMean(D(n),D(n-
1),.5))/(((Rp(n)).^2).*Axw(n)));
        ap0(n)= Ax(n)./dt;
        ap(n) = ae(n) + aw(n) + ap0(n) - Sp(n)*Ax(n);
        b(n)  = ap0(n)*Colast(n) + Sc(n)*Ax(n) + J;

    otherwise
        error('Boundary %s not recognized',varargin{ind+1});
end

% Check for negative coefficients
if (any(ae)<0 || any(aw)<0 || any(ap)<0 || any(b)<0), keyboard; end

% Solver
Co = TDMA(ap,aw,ae,b,Co0,n);
iter = 1; err = 0;
end

function xi = HarmonicMean(xP,xI,f)

xi = ( (1-f)./xP + f./xI ).^(-1);

end
```