ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

GRADO EN INGENIERÍA ELECTROMECÁNICA

DEVELOPMENT OF THERMAL FINITE ELEMENT MODEL OF AN ENERGY STORAGE DEVICE

Author: Valentín-Javier Sanz Ilundain
Director: Dimitri Torregrossa

Madrid
July 2015

Proyecto realizado por el alumno/a:
VALENTÍN-JAVIER SANZ IRUNDAIN
(Poner el nombre del alumno/a)

Fdo.: ..... Fecha: 07 / 07 / 2015

Autorizada la entrega del proyecto cuya información no es de carácter confidencial

EL DIRECTOR DEL PROYECTO
DIMITRI TORREGROSSA
(poner el nombre del Director del Proyecto)

Fdo.: ..... Fecha: 07 / 07 / 2015

Vº Bº del Coordinador de Proyectos

(poner el nombre del Coordinador de Proyectos)

Fdo.: ..... Fecha: ...../ ...../ .....

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN ACCESO *ABIERTO* (*RESTRINGIDO*) DE DOCUMENTACIÓN

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Valentín-Javier Sanz Ilundain, como alumno de la UNIVERSIDAD PONTIFICIA COMILLAS (COMILLAS), **DECLARA**

que es el titular de los derechos de propiedad intelectual, objeto de la presente cesión, en relación con el Proyecto de Fin de Grado titulado Development of thermal finite element model of an energy storage device, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual como titular único o cotitular de la obra.

En caso de ser cotitular, el autor (firmante) declara asimismo que cuenta con el consentimiento de los restantes titulares para hacer la presente cesión. En caso de previa cesión a terceros de derechos de explotación de la obra, el autor declara que tiene la oportuna autorización de dichos titulares de derechos a los fines de esta cesión o bien que retiene la facultad de ceder estos derechos en la forma prevista en la presente cesión y así lo acredita.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad y hacer posible su utilización de *forma libre y gratuita* (*con las limitaciones que más adelante se detallan*) por todos los usuarios del repositorio y del portal e-ciencia, el autor **CEDE** a la Universidad Pontificia Comillas de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución, de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra (a) del apartado siguiente.

3º. Condiciones de la cesión.

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia, el repositorio institucional podrá:

(a) Transformarla para adaptarla a cualquier tecnología susceptible de incorporarla a internet; realizar adaptaciones para hacer posible la utilización de la obra en formatos electrónicos, así como incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.

(b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato. .

(c) Comunicarla y ponerla a disposición del público a través de un archivo abierto institucional, accesible de modo libre y gratuito a través de internet.[1]

(d) Distribuir copias electrónicas de la obra a los usuarios en un soporte digital. [2]

4º. Derechos del autor.

El autor, en tanto que titular de una obra que cede con carácter no exclusivo a la Universidad por medio de su registro en el Repositorio Institucional tiene derecho a:

a) A que la Universidad identifique claramente su nombre como el autor o propietario de los derechos del documento.

b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.

c) Solicitar la retirada de la obra del repositorio por causa justificada. A tal fin deberá ponerse en contacto con el vicerrector/a de investigación (curiarte@rec.upcomillas.es).

d) Autorizar expresamente a COMILLAS para, en su caso, realizar los trámites necesarios para la obtención del ISBN.

d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

[1] En el supuesto de que el autor opte por el acceso restringido, este apartado quedaría redactado en los siguientes términos:
(c) Comunicarla y ponerla a disposición del público a través de un archivo institucional, accesible de modo restringido, en los términos previstos en el Reglamento del Repositorio Institucional

[2] En el supuesto de que el autor opte por el acceso restringido, este apartado quedaría eliminado.

a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.

b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.

c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.

d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

a) Deberes del repositorio Institucional:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.

- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.

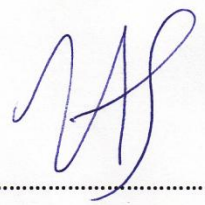
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.

b) Derechos que se reserva el Repositorio institucional respecto de las obras en él registradas:

- retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 20 de JULIO de 2015

ACEPTA



Fdo.....

UNIVERSIDAD PONTIFICIA DE COMILLAS
ESCUELA TECNICA SUPERIOR DE INGENIERIA (ICAI)
GRADO EN INGENIERIA ELECTROMECANICA

DEVELOPMENT OF THERMAL FINITE ELEMENT MODEL OF AN ENERGY STORAGE DEVICE

Autor: Sanz Ilundain, Valentín-Javier.

Director: Torregrossa, Dimitri.

Entidad Colaboradora: École Polytechnique Fédérale de Lausanne (EPFL)

RESUMEN DEL PROYECTO

Introducción

Con el paso del tiempo, las redes de transmisión de energía eléctrica parecen cada vez más obsoletas. Además de la creciente escasez de las fuentes convencionales de electricidad, la estructura actual de la red eléctrica no es capaz de integrar eficazmente nuevas fuentes de energía renovable como la solar o la eólica. Por el contrario, las microrredes inteligentes se desarrollan cada vez más visto que ofrecen una buena integración de dichas energías renovables en sus estructuras y suponen una solución a los problemas arriba mencionados.

El almacenamiento de la energía es un aspecto clave de estas microrredes y por tanto nuevos y más eficientes sistemas de almacenamiento están siendo desarrollados.

Las baterías Li-ion están conociendo un increíble crecimiento de popularidad debido a sus muy deseables características como su alta densidad energética o su excelente característica de carga-descarga que les permite operar más de 500 ciclos sin perder capacidad. Además, el boom de los dispositivos electrónicos portátiles, equipados la mayoría de las veces con este tipo de baterías, ha incrementado el interés en esta tecnología y cada vez más recursos son invertidos en investigar y desarrollar la familia de batería ion litio cada año.

Este proyecto se inscribe a esta corriente y se dedica a estudiar el comportamiento térmico de una batería Li-ion.

Metodología

La intención de este proyecto es crear un modelo térmico de elementos finitos de la batería considerada en el que poder correr ciertas simulaciones. Varias de ellas son lanzadas y el comportamiento del modelo frente a distintos inputs es observado. Aparte, se pretende construir un sistema basado en los resultados de dichas simulaciones y en el comportamiento del modelo que sea capaz de predecir la respuesta térmica del modelo de la batería para cualquier entrada.

El modelo térmico de elementos finitos ha sido creado usando el software ANSYS. Estos modelos dividen el objeto estudiado en pequeñas áreas y/o volúmenes: los llamados "elementos finitos". Cada uno de estos elementos se asocia con un cierto número de puntos denominados nodos. Este número varía en función del elemento asociado y su forma. De esta forma, el problema se simplifica: en lugar de resolver el problema para un número infinito de puntos, se resuelve únicamente en los nodos (número finito). Una vez las soluciones nodales obtenidas, es posible saber la solución en cualquier otro punto mediante interpolación.

Para este caso, se han tenido en cuenta dos modelos distintos:

- Uno "real" que reproduce con gran exactitud la batería real. Considerando las características de esta, el modelo será excesivamente pesado: está conformado por un gran número de pequeñísimas capas y tendrá por lo tanto un número enorme de elementos finitos y de nodos. El programa tardaría demasiado en resolver este sistema.
- Uno "aproximado", mucho más útil y manejable. En este caso, las distintas capas de la batería se agrupan según su material. El modelo resultante tiene así un número reducido de capaz de tamaño mediano y por lo tanto, menos elementos finitos y nodos que el otro modelo. Se alcanza la solución mucho más rápidamente con este modelo y por tanto, se usará este para la realización del trabajo.

Una vez que la geometría del modelo ha sido definida y las propiedades de los materiales asignadas a cada capa del modelo, se lanzan distintas simulaciones. Las diferentes entradas de las simulaciones modelan las pérdidas térmicas en la batería y tienen distintas amplitudes y duraciones. Incluso si el modelo utilizado es más ligero y sencillo, los tiempos de cálculos son muy largos. Para reducirlos, el paso de tiempo se fija a 5 segundos entre valor y valor: el tiempo de simulación se reduce así por cinco y en estas condiciones, cada experimento dura menos de un día. En todas las simulaciones, el modelo alcanza el régimen permanente a los 60000 segundos de su inicio. Entre todas las simulaciones, los resultados de una en particular son especialmente importantes como se muestra a continuación: se trata de la respuesta del sistema a una entrada de 1W durante 30s.

Una vez el análisis térmico en ANSYS terminado y sus resultados almacenados, toca construir el sistema de predicción del comportamiento térmico del modelo. Se basa en el teorema de la convolución y en una propiedad en particular que dice que la transformada de Laplace de la convolución de dos funciones es el producto de las transformadas de Laplace de cada una de dichas funciones. Considerando también la propiedad de que la transformada de Laplace de la delta de Dirac es igual a la unidad, la función de transferencia del sistema será igual a su respuesta térmica si la entrada es una delta de Dirac.

Considerando que 30 segundos son despreciables frente a los 60000 segundos de tiempo total de simulaciones, se puede aproximar un input de 1 vatio durante 30

segundos a una delta de Dirac. Por lo tanto, la respuesta del sistema ante esta entrada será equivalente a la función de transferencia en el dominio del tiempo. En consecuencia, el comportamiento de la batería puede predecirse usando este sistema, resumido en la siguiente ecuación:

$$y(t) = y_{1W,30s}(t) * x(t)$$

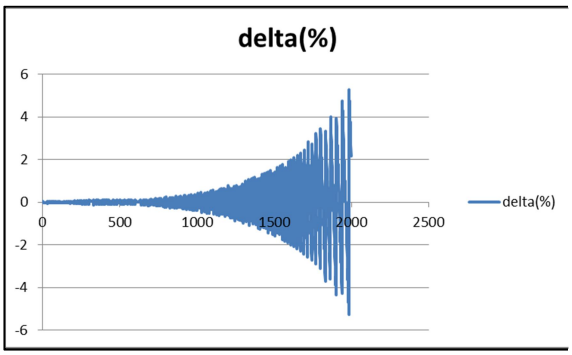
Esta ecuación ha de calcularse para todos los nodos considerados y para todas las simulaciones. Esta es una tarea tediosa que llevaría mucho tiempo. Se puede resolver este problema diseñando una serie de códigos en Matlab que calculen automáticamente el producto de convolución para cada nodo para la entrada deseada y que luego lo comparen con los resultados obtenidos con el análisis térmico de ANSYS. Tanto los datos de salida como los de entrada de estos códigos se almacenan en un archivo Excel que hace las veces de coordinador entre estos códigos.

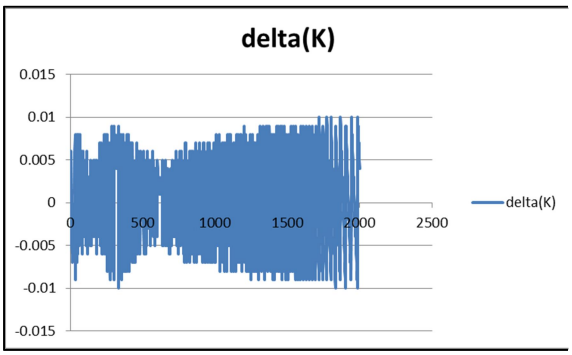
Para aproximar la entrada de 1W durante 30 segundos a una delta de Dirac, es necesario realizar un muestro de la entrada y de la función de transferencia del sistema. Mediante esta operación, el paso de tiempo entre valores cambia de 5 a 30 segundos y por tanto las entradas han de ser múltiplos de 30.

Resultados

Los resultados obtenidos usando ambos métodos son comparados analíticamente usando Excel y gráficamente con un script de Matlab.

El archivo Excel almacena los valores obtenidos usando ambos métodos y los compara calculando sus errores absoluto y relativo.

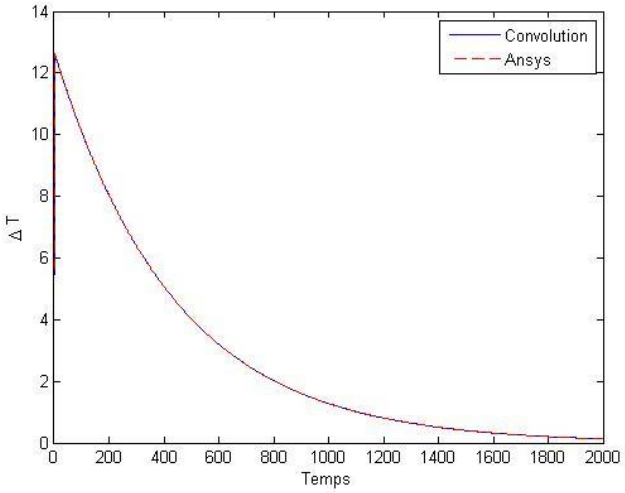




El error absoluto está comprendido en todo momento en un rango de -0.01 a 0.01 grados Kelvin durante toda la simulación. El error relativo es muy bajo excepto para los instantes finales de la simulación, cuando aumenta rápidamente aunque permaneciendo en todo momento en un rango de error del 5%. Este crecimiento se debe a que mientras que la temperatura de la batería disminuye en todo momento, el error absoluto se mantiene en los mismos valores.

Utilizando un código de Matlab, la gráfica de los resultados de ANSYS se puede

comparar con la de los obtenidos usando el producto de convolución.



Las dos gráficas se superponen: prácticamente no hay diferencia entre los resultados obtenidos con ambos métodos.

Conclusiones

Como ya se mencionó, un modelo de elementos finitos de una batería de ion litio ha sido construido. En base a los resultados obtenidos simulando diferentes cargas térmicas en el modelo, se ha podido desarrollar un sistema capaz de predecir la respuesta de la batería a distintas entradas.

Las gráficas presentadas anteriormente muestran que las predicciones de la respuesta del modelo de la batería obtenidas aplicando el teorema de la convolución para una entrada dada son extremadamente precisas pues no hay apenas diferencia entre los resultados obtenidos con uno y otro método. Por lo tanto, se puede concluir que el sistema de predicción basado en el producto de convolución es correcto.

Una vez que los modelos de elementos finitos y de predicción han sido construidos y teóricamente confirmados, es el momento de usarlos en aplicaciones reales. Trabajos futuros deberán comparar el comportamiento térmico real de una batería para una entrada conocida con los resultados del modelo. De esta forma, la validez de las aproximaciones realizadas para construir el modelo puede ser confirmada o desestimada.

Una vez el modelo térmico de elementos finitos haya sido verificado experimentalmente, debería ser posible de predecir el comportamiento de una batería Li-ion real utilizando el teorema de la convolución y conociendo previamente su respuesta a una entrada aproximadamente igual a una delta de Dirac, lo que a su vez serviría para confirmar este sistema. Usando este método, el análisis de cualquier

sistema de almacenamiento de energía se simplifica mucho puesto que sólo se requieren los resultados de una única simulación para poder realizar el producto de convolución.

SUMMARY OF THE PROJECT

Introduction

As time goes by, the conventional electric power grids seem more and more obsolete. With the conventional electrical power sources becoming scarcer, its actual structure fails to efficiently integrate new renewable power sources such as the solar or the wind power. In opposite, smart microgrids are being increasingly developed since they offer a good integration of renewable energies and they are a solution to the above mentioned problems.

Energy storage is a key part of these microgrids and therefore, new and more efficient energy storage systems are being researched and developed.

Li-ion batteries are experiencing an incredible popularity growth due to its very desirable characteristics like its high energy density or its excellent charge-discharge characteristic which allows them to operate more than 500 cycles without losing capacity. Added to this, the boom of the portable electronic devices which most of times are equipped with these batteries has increased the interest on this technology and more and more resources are invested on researching and developing the Li-ion family every year.

This project follows this trend and dedicates to studying the thermal behavior of a Li-ion battery.

Methodology

The intention of this paper is to create a thermal finite elements model of the considered battery in which some simulations could be run. Several of them are performed and the model behavior to different inputs is stored. Additionally, it is intended to build a system based on the simulations results and the model behavior able to predict the thermal response of the battery model to any input.

The thermal finite elements model is created using the software ANSYS. These types of models divide the studied object into small areas and/or volumes: the “finite elements”. Each one of these elements is associated with a certain number of points called nodes. This number varies depending on the element and its form. This way, the problem is simplified: instead of solving the problem for an infinite number of points, it is only solved at the nodes (finite number). Once the nodal solutions are obtained, it is possible to know the solution at any point interpolating the nodal results.

For this case, two different models have been considered:

- A “real” one, which reproduces exactly the real battery. Given the considered battery characteristics, this model is way too heavy to be useful: it is made of a huge

number of very small layers and therefore, it has an unmanageable number of finite elements and nodes. The software takes too much time to obtain the solution.

- An "approximate" one, more useful and manageable. Here, the different layers that shape the battery are summed up according to its material. The obtained model has a reduced number of medium sized layers and therefore, less finite elements and nodes than the other model. The solution is reached much faster with this model and therefore, it is the one used in the project.

Once the geometry of the model is defined and the materials properties are assigned to each layer of the model, the different simulations are launched. The different inputs of the simulations model the heat losses of the battery and have different amplitudes and durations. Even if the approximate model is lighter and simpler, the calculations take too long. In order to reduce the calculation time, the time step is set to 5 seconds between value and value: the simulation time is then reduced five times and with these conditions, each experiment lasts less than a day. In all the simulations, the model reaches steady-state conditions at a time of 60000 seconds. Amongst all the simulations, the results of one in particular are especially important as it is shown next: the response of the system for an input of 1W during 30 seconds.

Once the thermal analysis calculations on ANSYS are done and their results stored, it is time to build the prediction system of the model thermal behavior. It is based on the convolution theorem and in a particular property that states that the Laplace transform of the convolution of two functions is the product of the Laplace transforms of each of these functions. Considering as well the property that the Laplace transform of the Dirac delta function is the unit, the transfer function of the system is equal to its thermal response if the input is a Dirac delta function.

Given that 30s is almost negligible versus the total time of 60000s, it can be considered that an input of 1 Watt during 30 seconds is an approximation of a Dirac delta function. Therefore, the response of the system to this input is its transfer function in the time domain. Consequently, the battery behavior to any input can be predicted using this system, summarized in the following equation:

$$y(t) = y_{1W,30s}(t) * x(t)$$

This equation has to be calculated for every considered node and for every simulation. This is a rather tiresome task that would take too much time to accomplish. The problem is solved with the design of a set of Matlab codes that calculate automatically the convolution for every node for a desired input and then compare its results with the ones obtained with the thermal analysis performed on ANSYS. Both inputs and outputs of these codes are stored into an Excel file that serves double duty: it stores the results from ANSYS and from Matlab and it connects the different codes.

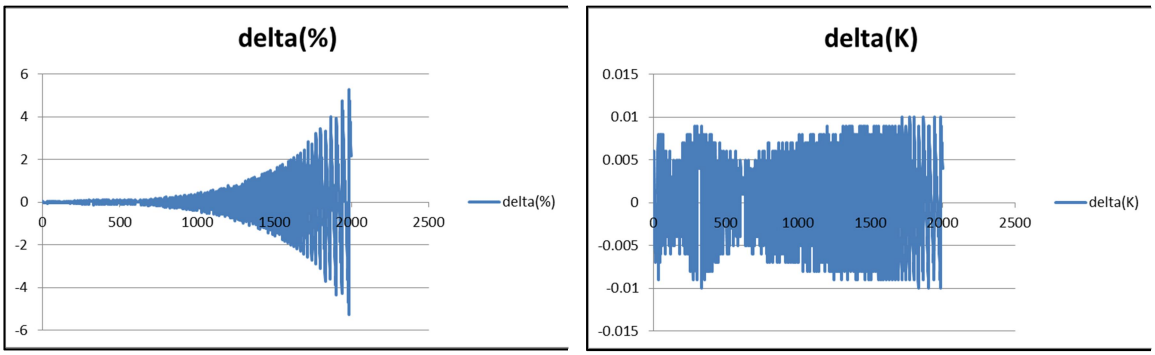
In order to approximate the input of 1W during 30s to a Dirac delta function, it is necessary to sample both the input and the system transfer function (in other words, the

response of the system to the 1W, 30s input). With this operation, the time step between values changes from 5 to 30 seconds and thus the inputs have to be multiples of 30.

Results

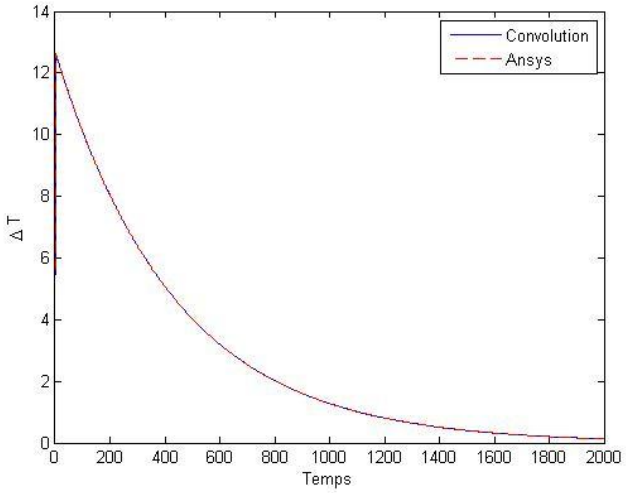
The results obtained using both methods are compared analytically using Excel and graphically with a Matlab script.

The Excel file stores the values obtained using both methods and compares them calculating their relative and absolute errors.



The absolute error stays within a range of -0.01 and 0.01 during all the simulation. The relative error is very low except for the last stages, where it skyrockets. However it doesn't go over a 5% error. The growth of the relative error is due to the fact that the temperature of the battery decrease at all times while the absolute error stays the same during all the simulation time.

Using a Matlab code, the graph of the results obtained with the convolution can be displayed simultaneously with the one of the results from ANSYS.



The two graphs are overlapping: there is almost no difference between the results obtained with both methods.

Conclusions

As it has been mentioned, a finite elements model of a Li-ion battery has been built. Based on the results obtained by simulating different thermal loads on the model, it has been possible to develop a system able to predict the response of the battery to different inputs.

The graphs presented above show that the predictions of the response of the battery model obtained applying the convolution theorem for a given input are extremely accurate since there is almost no difference between the results obtained with both methods. Therefore, the forecasting system based on the convolution theorem is correct.

Once the finite elements and forecast models are built and theoretically confirmed as correct, it is time to use them in real applications. Further work should compare the thermal behavior of a real battery for a given input to the one from the model. This way, the validity of the approximations made to build the model can be confirmed.

Once the thermal finite elements model is verified, it should be possible to predict the behavior of a real Li-ion battery using the convolution theorem and previously knowing its response to a Dirac delta function like input, which should confirm the validity of this system as well. The analysis of any energy storage system would therefore much simpler using this method since it only requires the results of one single simulation to calculate the convolution.

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

GRADO EN INGENIERÍA ELECTROMECÁNICA

DEVELOPMENT OF THERMAL FINITE ELEMENT MODEL OF AN ENERGY STORAGE DEVICE

Author: Valentín-Javier Sanz Ilundain
Director: Dimitri Torregrossa

Madrid
July 2015

ABSTRACT

This paper presents a way of modeling an energy storage system such as an electrochemical cell (battery) using thermal finite elements models. Using the thermal analysis tools of the software ANSYS, the model has been used for different patterns of heat generation in the battery. From these results, the thermal response of the cell associated to different heat generation rates can be predicted by using the convolution theorem. In order to achieve this goal, several MATLAB files have been created, allowing for the calculation of the thermal response of the system from the results obtained with the thermal finite elements model. The inputs and the outputs of these programs are both read and written into Excel files, whose function is not only to store the data but to serve as a connection between the different .m codes.

From a scientific point of view, the use of convolution theorem for predict the temperature evolution associated to any arbitrary heat generations has been achieved in simulation environment.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

- ESS Energy Storage System
- FE Finite Elements
- Li-ion Lithium-ion
- LTO Lithium Titanate Oxide ($Li_4Ti_5O_{12}$)
- NCA Nickel Cobalt Aluminum Oxide
- PVDF Polyvinylidene Fluoride

1 Introduction

The aim of this project has been to create a thermal FE model of an energy storage system in order to predict its response (time evolution of the surface temperature in specific points) to any heat generation wave form by means of the convolution theorem. To do so, the structure of this document is summarized in the following lines.

First, the targeted energy storage system is defined as well as its characteristics, mainly the properties of the materials and an estimation of the heat losses. Then, the FE method is discussed in a general way and the main characteristic of the thermal analysis with ANSYS are exposed.

After these two starting sections, the model considered for the targeted ESS is presented in the fourth section along with some thermal simulations and results. In particular, the results for the impulsion of 1W are really useful for the next part.

Finally, the convolution theorem and how it is applied to the model are explained. Additionally, some considerations for integrating the convolution theorem to the model in the MATLAB programs are presented with some results.

The last part summarizes the conclusions of the paper and details the next steps to be undertaken as well as some possible applications.

In addition, some appendices can be found at the end of the document: appendix A corresponds to the code ‘Convolution.m’, which is the basic tool used for applying the convolution to the model. Additionally, appendix B is the code ‘HMap.m’ used to build heat maps from the results stored in the Excel. Finally, appendix C corresponds to the code ‘AnsysVsConv.m’ that compares the results obtained from the convolution with the ones from the thermal analysis performed with Ansys.

2 Energy Storage System: the Li-ion battery

The energy storage system considered in this paper is the Lithium-ion battery. Throughout this part, the history and state of the art of this technology are quickly resumed and some general information about its properties is given, focusing specially on its thermal losses. Following this, the geometry and the properties of the specific battery used for this project are also presented.

2.1 Overview

2.1.1 History

Gilbert N. Lewis, an American physical chemist, started studying the lithium battery technology in 1912. Despite this pioneering work, the technology wasn't developed seriously until the 1960s as lithium started being used as the anode of high-energy-density battery systems. The lithium batteries, non-aqueous and non-rechargeable, were first used in the 1970s for military applications. In the 1980s, some attempts were made to develop rechargeable lithium batteries but the project didn't achieve any good results due to the instability of the metallic lithium anode. Theoretically, this kind of rechargeable batteries could provide very high energy densities but unfortunately cycling produces some particles that penetrate the separator and thus produce an electrical short causing thermal runaway. This instability that characterizes the lithium metal changed the focus of the research to batteries using lithium ions in a non-metallic solution.

In 1991, Sony started commercializing the first Li-ion batteries. Although their energy density was lower, these batteries are safe if they are used within certain security range. Since 1994, the production cost has dropped continuously while the specific energy of this type of battery grows non-stop. These virtues added to the absence of toxicity gives a great advantage to the Li-ion battery over other battery technologies.

In 2009, 38 percent of all batteries were Li-ion, being specially used in portable applications (laptops, cell phones…) and more and more used in heavier applications (electric vehicles). In 2011, the 66% of the Japanese rechargeable batteries were Li-ion. This incredible growth is caused to the desirable characteristics of this type of batteries and, in part, because of the boom of the market for portable electronic devices, which fancy this type of batteries.

The research and development of this technology continues as its popularity growth doesn't seem to stop. In 2015, the company Tesla Motors launched "Tesla Powerwall", a rechargeable Li-ion battery aiming to store electricity from renewable sources for domestic consumption and proposing an alternative to the traditional power grids.

2.1.2 Characteristics and properties

A simple Li-ion battery consists on a cathode (metal oxide), an anode (carbon) and electrolyte as conductor. The current is created by the flow of lithium ions from the cathode to the anode via the electrolyte if the battery is discharging. This movement causes oxidation at the anode since it loses electrons as the positive ions arrive. Consequently, the discharge of the battery causes as well a reduction phenomenon at the cathode. During charge, the movement of the lithium ions goes in the opposite direction and as a consequence the cathode oxidizes as the anode undergoes a reduction.

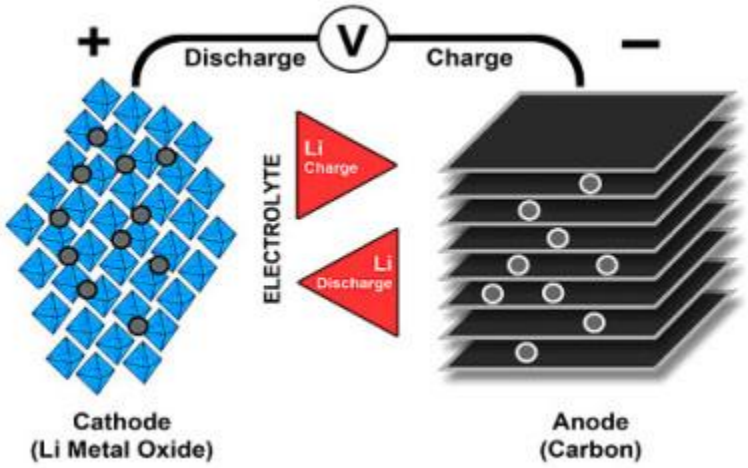


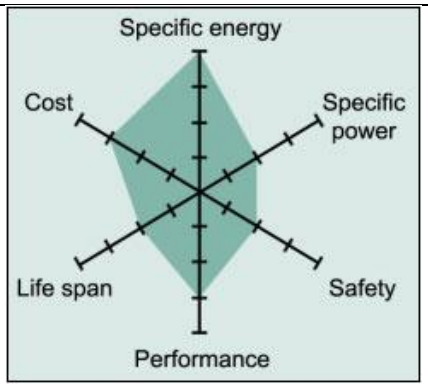
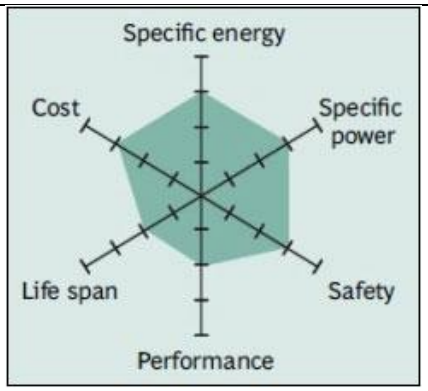
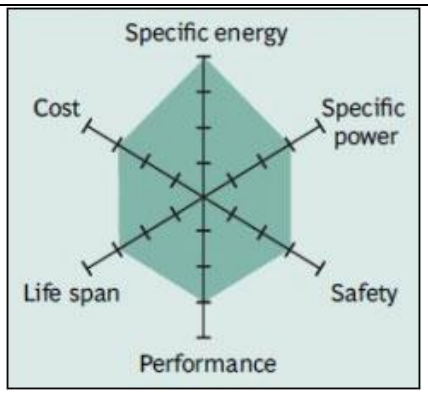
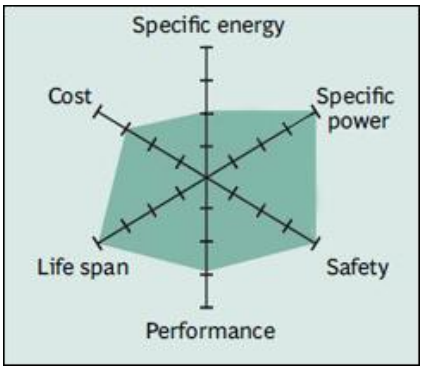
Figure 1 - Working principle of Li-ion batteries.

There exist various types of Li-ion batteries. The main different categories existing correspond to the various types of cathodes that can be chosen but it is still possible to vary the election of cathodes. Depending on the election of these two components, the characteristics of the battery vary.

The main categories for the cathodes and their characteristics are summarized next:

- Lithium Cobalt Oxide ($LiCoO_2$): it has a very high capacity but a low specific power: it cannot be charged nor discharged at a current higher than its C-rate; forcing a fast charge causes overheating and the chances of burning dramatically increase.
- Lithium Manganese Oxide ($LiMn_2O_4$): It has very high power (currents up to 10C or 30C for a pulse) but its capacity is low.
- Lithium Nickel Manganese Cobalt Oxide (NMC): High capacity and high power (up to 2C) and a long cycle life (up to 2000).
- Lithium Iron Phosphate ($LiFePo_4$): Very low capacity with very high power (up to 25C) with a long cycle life (up to 2000). Very safe technology.
- Lithium Nickel Cobalt Aluminum Oxide (NCA): Very high capacity and quite high power (allows fast charge but the discharge is limited to 1C).

The properties of these different batteries are summarized in the following table:

$LiCoO_2$	
$LiMn_2O_4$	
NMC	
$LiFePo_4$	

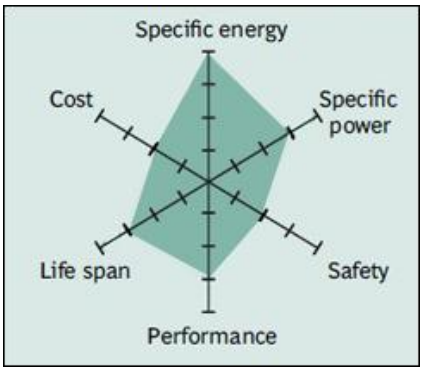
NCA	

Table 1 - Summary of the different Li-ion batteries properties.

Since the initial state of development of the Li-ion battery it is known that the preferred cathode is carbon since it is a cheap material of low mass, very available, and it is able to take up a large quantity of lithium so that the cell voltage does not vary too much between the fully charged state and the discharge. There are many carbon sources useful for building the battery. The original Li-ion battery created by Sony in 1991 used a negative electrode made of coke (fuel with high carbon content made from coal or oil). However, the different sources of carbon vary in their ability to intercalate lithium ions: in particular, the coke is able to take up half as much ions as graphite. In the present, most Li-ion batteries employ graphite electrodes as cathode. Other sources of carbon being more and more used nowadays are pyrolysed polymers as the polyvinylidene fluoride (PVDF), normally combined with graphite.

Another alternative for the anode is using a lithium titanate electrode (Li4Ti5O12 or LTO). This type of anode is normally combined with a graphite cathode and the resultant Li-ion battery has a low nominal voltage and low capacity. However it has a high specific power (10C for continuous use), allows fast charge, it is a totally safe technology and has the longest lifetime: it is able to perform up to 20000 cycles. The absence of the carbon anode allows for achieving this long lifetime. The carbon anodes produces Solid electrolyte interface (SEI) that is the main source of ageing in classical Li-ion cells. The general attributes and advantages of the Li-ion technology are summarized next:

- High energy density.
- High operating voltage.
- Excellent charge-discharge characteristics: able to operate more than 500 cycles without losing capacity.
- Long lifetime.
- Much safer than a metallic lithium battery.
- Rapid recharging possible (most cases).

2.1.3 Thermal losses

The aim of the project is to model an energy storage device and analyze its thermal response to a heat generation impulse. The source of this impulse is the thermal losses of the battery itself considering that the battery workplace is at room temperature and there are no external heat sources.

Two main types of thermal losses can be identified: the first one is associated with the Joule effect (irreversible) and a second one produce at the interface between the electrolyte and the electrodes with thermodynamic origin.

The Joule losses in the battery follow the equation:

$$P_{joule} = i^2 \cdot R_{interne} = i \cdot U_R \quad , with\ U_R = U_{eq} - U \tag{2.1}$$

For low loads (the battery working below its nominal conditions) are almost negligible. When the battery works at least at nominal conditions (discharge rate of 1C or superior), the losses due to the Joule effect are more important than the thermodynamic reaction losses. The term of the current in the equation being squared, any variation of this parameter has a deep impact in the final heat generation.

The source of the thermodynamic losses, or heat reaction losses, is the entropy variation at the electrodes. Therefore, it depends on the type of electrodes used in the considered battery. The reaction may be exothermic or endothermic: for the charge, reaction is at first endothermic then it becomes slightly exothermic. In the case of the discharge it is the opposite case. It is possible to find graphs describing the entropy variation at the electrodes function of the state of charge in literature.

Some equations modelling the whole system can be written. From a thermodynamic point of view, the heat generated by the system would be:

$$Q = I \cdot [(U_{OCV} - U) + T \cdot \frac{\delta U_{OCV}}{\delta T} \tag{2.2}$$

This equation highlights the importance of the battery equilibrium voltage and its temperature derivative. Both heat sources specified above can be found in that equation, with the difference of voltages corresponding to the Joule effect and the temperature derivative to the heat reaction.

2.2 Geometry of the battery

The physical battery studied in this paper is a Li-ion battery combining several of the technologies mentioned above. The anode is a combination of LTO, conductive carbon and pyrolysed polymers while the cathode is also made of carbon, pyrolysed polymers and NCA as active material. The collectors for the anode and the cathode are made of copper and aluminum respectively. The last element is the separator, made of aluminum

oxide (Al_2O_3) and pyrolysed polymer.

The battery has a squared shape and its internal structure is formed by layers of the elements mentioned in the paragraph above. The layers of materials are disposed following a pattern that is repeated several times. Therefore, this pattern is symmetric and it is formed by at least one layer of separator, anode, cathode and their collectors.

The thickness of the elementary pattern is the sum of the thicknesses of the different layers shaping it and as a consequence, the total thickness of the battery is the one of the pattern multiplied by the number of repetitions.

In order to build the model and perform a thermal analysis, ANSYS needs some properties of the material layers: the density, the conduction coefficient and the heat capacity (specific heat). Some hypotheses have to be made at this point.

The properties of the aluminum oxide have been considered for a 96% of oxide [4]. The chosen pyrolysed polymer is PVDF [5] and for the conductive carbon, the chosen material is graphite. Finally, it has been considered that the conduction coefficients of the materials do not vary with the direction of the heat flow ($K_{xx}=K_{yy}=K_{zz}$). The properties of the materials are summarized next:

Material	Density [kg/m^3]	Conduction coefficient [W/m-K]	Heat capacity [J/kg-K]
Al	2700	237	902
Cu	8900	398	385
Al_2O_3	3720	25	880
PVDF	1780	0,2	1400
Graphite	2100	25	710
LTO [6]	3510	1,04	1437
NCA [7],[8]	1240	2,49	601

Table 2 - Properties of the battery materials

From the properties exposed in table 2 it is possible to obtain the properties of the 5 different layers of the elementary pattern.

3 The Finite Element Method

The model of the Li-ion battery studied in this paper is done using the FE method. The method itself is briefly introduced in first place and then the specific software, ANSYS, used to build the FE model is presented focusing specially in its thermal analysis tools.

3.1 General overview

3.1.1 History

The FE method was initially developed by the German mathematician Richard Courant in 1943 who tried to solve a vibrating system by numerical analysis and minimizing the calculus variables. When the first computers arrived in the 1950s, the dominating methods for solving problems were iterative and done manually. Though these methods were simpler than solving the gigantic equations associated with the alternate ones, they still were heavy and some problems took even several weeks of calculations in order to be solved. The arrival of the computers allowed changing to the other methods, which equations were known for a long time but were impossible to solve due to its complexity. An example is the Navier-Stokes equations which describe the movement of a fluid.

In the following decades, the FE method's popularity grew considerably amongst the industry and the academic community. In the 1970s, the number of papers studying the method grew and it started being used to solve a great variety of problems. During this decade, the FE method requirements limited its use to powerful and expensive computers owned by important industries, being not accessible to the general public.

From the 1980s onwards and thanks to the popularization of the personal computers, the use of commercial and specialized programs spread, including pre and post-processors allowing the representation graphic of the model, the mesh and the results. Nowadays, with the incredible growth of the computers capacity, the FE method has obtained a great accuracy.

3.1.2 Introduction to the method

The FE method is a numerical one. Its goal is to obtain the field variables, the unknowns of the problem. Once these variables are known, the whole system can be described. These variables are displacements in solid mechanics, velocities in fluid mechanics, potentials in electric and magnetic problems and temperatures in heat flow problems, the case studied.

The method simplifies a problem by dividing the solution region into small parts (finite elements). Doing this the number of unknowns is finite instead of infinite (continuous

case). The field variables are expressed in terms of approximating functions within each element. The functions are defined in terms of field variables of points called nodes. So as a summary, the unknown field variables of the system are written in function of the field variables of the nodes. The goal of the FE method is then to solve and find these nodal variables. Once they are known, the field variables of any point can be found interpolating the nodal solutions. Thus, the found solutions are approximations except for the nodes, only points where the solution is exact. The set of points where the solution is exact, that is the set of nodes, form a mesh. The areas defined by the lines of the mesh are the finite elements. This is illustrated below.

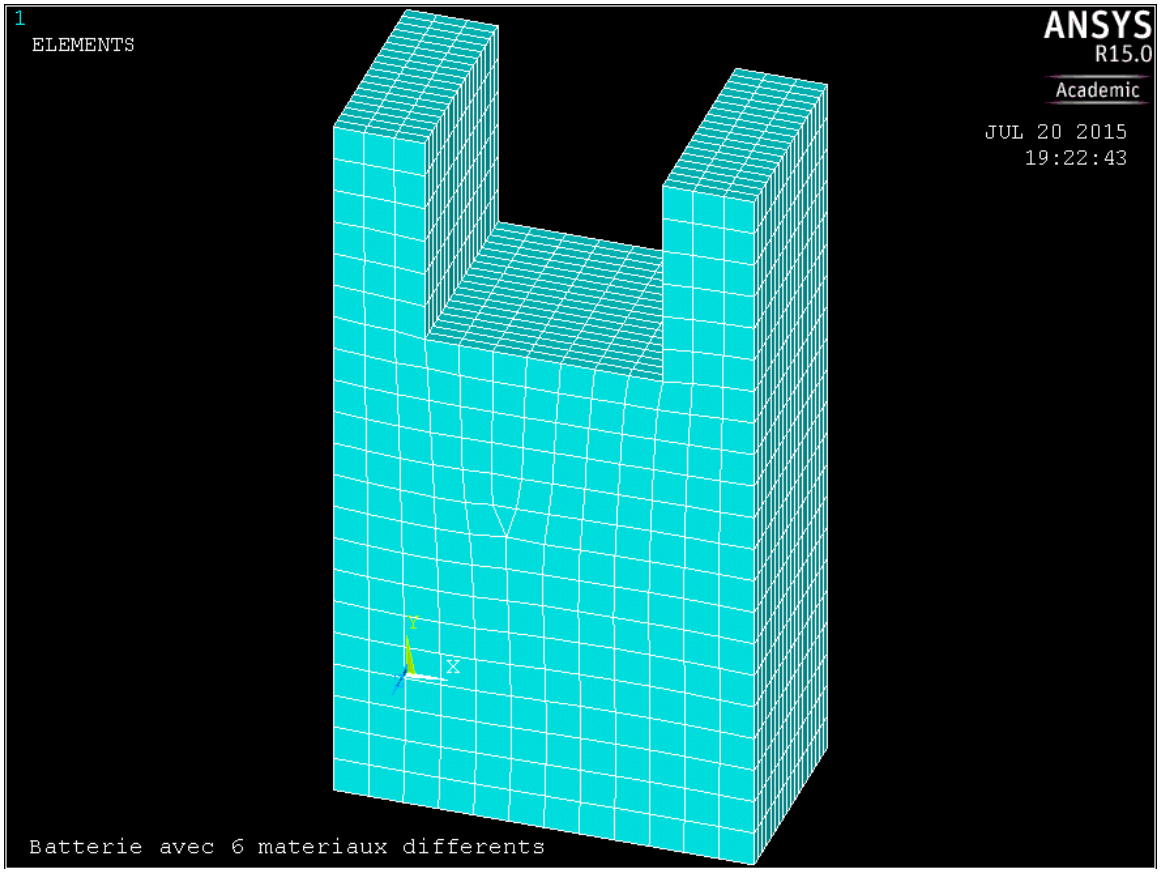


Figure 2 - Meshed FE model.

In figure 2, an example of FE model is presented. The white lines scarring the model form the mesh. Each and every one of the intersections is a node and the portions of model secluded by the mesh are the elements. Sometimes, normally when the object is irregular like the one in the example, the mesh can't keep a regular shape and some elements have irregular forms. It can be seen in figure 2 at the element with triangular shape.

Therefore, the choice of the right type of element is very important when using FE method to solve a problem. Some properties have to be assigned to the elements and they may vary depending on the problem at hand. For example, the elements used in this project for the thermal analysis of the Li-ion battery have the temperature as sole

degree of freedom at their nodes.

Other properties have to be taken into consideration while choosing the type of elements used in the model: its dimension (2D or 3D), its geometrical shape and the number of nodes associated to the element. For the model presented in this paper, the used elements and its characteristics are discussed in part 3.2.

An analysis under the FE method is divided in three stages: pre-processing, calculus or solver and post-processing.

During the pre-processing stage, the elements and materials of the model as well as the geometry are defined. The mesh is generated dividing the continuous model into finite elements with their respective nodes. The contour conditions (the room temperature for example) are defined as well as the desired input and its source. The conditions of the simulation are defined too such as the final time or the time step.

At the calculation stage, the system nodes unknowns are solved. The calculation time depends on a lot of different factors such as the number of nodes in the system, the total simulation time or even the type of problem to be solved.

Finally, once the calculation is over, the results are extracted and treated at the post-processing stage. They can be represented in a graphical way or simply exported to other data processing software, as is explained in further paragraphs.

3.2 Thermal Analysis with Ansys

The basis for the thermal analysis in Ansys is a heat balance equation obtained from the principle of conservation of energy.

$$\rho c \left(\frac{\delta T}{\delta} + \{v\}^T\{L\}T\right) + \{L\}^T\{q\} = \ddot{q} \tag{3.1}$$

With:
ρ = density (command DENS)
c = specific heat (command C)
T = temperature (field variable)
t = time
{v} = velocity vector for mass transport of heat
{L}T = grad operator
$\{L\}^T$ = divergence operator

As stated before, choosing the elements used in the model is a crucial decision while using the FE method. Ansys provides a library including about 40 elements. They all have in common to have temperature as a degree of freedom at its nodes but the other properties are variable.

Selection criterions are the type of heat transfer (conduction/convection or radiation), if the problem is coupled (other field variables apart from the temperature are considered) or not and the shape, dimension and number of nodes of the element.

Ansys is able to solve problems involving the three primary modes of heat transfer: conduction, convection and radiation. The conduction is always considered when the model is built with thermal finite elements. For the convection, a surface load has to be applied to the conductive elements specifying the convection film coefficient and the bulk fluid temperature. Then, Ansys calculates the heat transfer across the surface with the load.

The radiation problems are non-linear but still solvable with Ansys. It can be defined using a special radiation link element, using surface effect elements specifying the radiation option, generating a radiation matrix or using the method Radiosity Solver.

Added to this, Ansys supports two different types of thermal analysis:

- Steady-state thermal analysis, which solves for the nodal temperature under steady-state heat load conditions: the heat storage effects varying in function of time can be ignored.
- Transient thermal analysis, which solves for the nodal temperatures under conditions that vary over a period of time. In this case, the heat generation has to be defined in the pre-processing stage in function of time.

Ansys programs are able to work in a variety of different unit systems. However, in this paper only the International System (SI) is considered. The units characterizing this system are shown in table 3.

Mass	$[kg]$	Density	$\frac{[kg]}{[m]^3}$
Length	$[m]$	Energy	$[J]$
Time	$[s]$	Heat Flow	$[W]$
Temperature	$[K]$	Heat Flux	$\frac{[W]}{[m]^2}$
Velocity	$\frac{[m]}{[s]}$	Film Coefficient	$\frac{[W]}{[m]^2[K]}$
Acceleration	$\frac{[m]}{[s]^2}$	Heat Generation	$\frac{[W]}{[m]^3}$
Force	$[N]$	Conductivity	$\frac{[W]}{[m][K]}$
Moment	$[N][m]$	Specific Heat	$\frac{[J]}{[kg][K]}$
Pressure	$[Pa]$	Dynamic Viscosity	$[Pa][s]$

Table 3 - Units in Ansys

The characteristics of the materials and the heat generation rates have to been written in a coherent way with the Table 3 written above.
More specific details about the geometry creation and the model definition as well as the concrete commands used are given in the chapter concerning the FE model itself.

4 The Battery FE model

In this chapter, the FE model of the battery is presented. As stated previously, it is made using ANSYS software. The tools and commands used to accomplish this task are presented in sections 1 and 2: first, the geometry creation is explained along with the hypothesis and the choices made and next, the part of the code in charge of defining the thermal conditions as well as performing the calculation itself is presented. Section 3 introduces the post-processing stage: what considerations have been taken into account in order to extract the results and how have them been exported.

4.1 Geometry creation

4.1.1 Element and material definition

Defining the elements and the materials used in the model is the first task of the pre-processing stage. Before doing anything, it is necessary to call the pre-processing database via the /PREP7 command. Once this is done, all the pre-processing commands that are described below are accessible. While doing this, the unit system has to be specified too. Since all the properties are written in International System units, the command '/UNITS, SI' is written immediately following the pre-processor call.

Two different elements have been considered for the model. Considering that the problem to be solved is purely thermal and as a consequence the only field variable that has to be considered is the temperature, there is no need to use coupled-field elements: their only degree of freedom is the temperature at their nodes. Added to this, the heat transfer in the battery is conductive (heat exchange between the layers) and convective (battery in contact with the air, at room temperature), the radiation has no influence whatsoever. Thus, the battery elements are pure and simple conductive elements (the convective load is applied at a later stage).

From the proposed library, two different elements are considered for the model. For the borders between layers and the external surfaces of the battery a 2-D element is required. The code calls PLANE55, which is a bi-dimensional quadrilateral element (4 nodal points corresponding to its vortex). Another one has to be considered though, since the model is three-dimensional and therefore requires an element for the internal layers. The chosen one is SOLID90, a brick with 20 nodal points. Both of them are associated with an element reference number by the means of the command ET.

Then, the different layers of the battery have to be defined along with their characteristics relevant to the studied problem. Using the command MP, the five layers are defined, associated to the properties explained in section 2.2 and given a material reference number. The defined materials and elements can be displayed on the model to see graphically its repartition which makes the error detection easier.

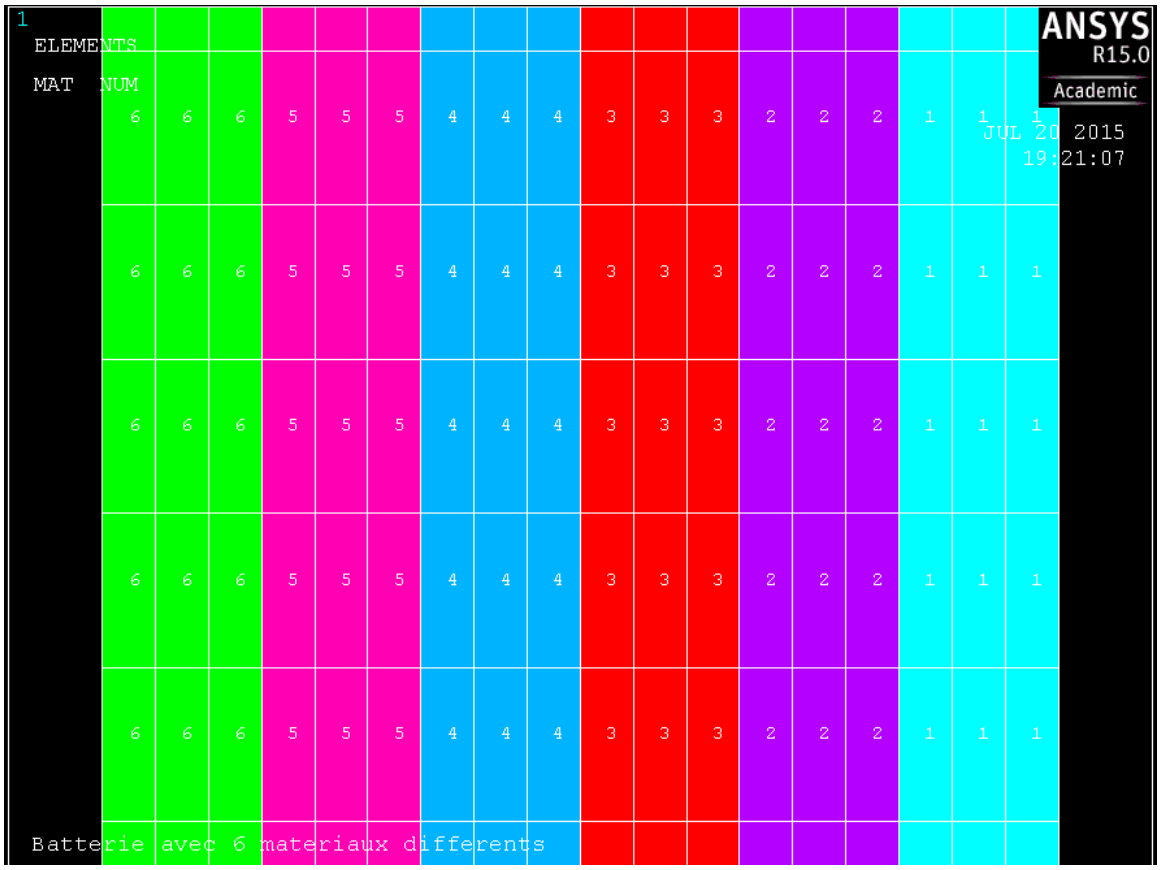


Figure 3 - Materials displayed over a FE model.

While building the geometry of the model, the elements and the material properties have to been associated to the created structures. To do so, the reference numbers previously allocated are used.

4.1.2 Real model

After defining elements and materials, the geometry itself has to be defined in a general way. The general dimensions of the battery are specified at this point as well as the thicknesses of each layer of material and the meshing size. The mesh is defined by two parameters: the length of each element in 2-D, specified while creating the mesh on a surface with the command LESIZE, and the number of divisions in 3-D, specified while performing an extrusion on a surface with the command EXTOPT. Note that the extruded surface has already been meshed bi-dimensionally and thus the 3-D mesh is completely defined. The mesh has to be defined carefully since Ansys impose some shape conditions that may be broken if the dimensions are too small.

The geometry has to be defined from the smallest structure to the biggest, from some points to the full volume. The first task is to create some key points and from this base some lines defining the borders of the object are created. The surface contained within these boundaries is then created, meshed and associated to an element type and a

material. The surface is extruded to generate a volume which is also meshed and given element and materials properties.

Following this procedure, the elementary pattern of layer for the Li-ion battery is created. In order to obtain the full object, VGEN is used. This command takes some pre-selected volumes and copies them into another point a specified number of times. The point is identified by a direction vector specified in the command. If the vector components are the pattern thickness for the extrusion direction and 0 for the other two directions and for the case of doing a single copy, the selected original volume is duplicated and the new created volume starts right where the original one ends, obtaining this way a single volume twice as big as the original one and following the same material pattern. Using this method and specifying the number of times that the pattern is repeated in the real battery the model of the real battery is created.

It is important to check for discontinuities that may be created in the model while using this method. These discontinuities present in the lines and areas limiting two different layers and are caused by item duplicities: some items may have been created several times. In order to correct these situations, the command NUMMRG is used: it merges equivalently defined items, for example two areas located at the same place and with the same element and material characteristics.

However, this model is not applicable in practice for several reasons. The different layers being extremely thin compared to the total thickness of the whole battery, the dimensions of the mesh model are very small and the shape conditions imposed by Ansys are violated. With these mesh dimensions, the number of finite elements in the model is very high and so is the number of nodes. Thus, the time required for solving the problem is unbearable: only the geometry creation takes around 20 minutes and the thermal analysis would take a much longer time, even if the field variables are calculated once every one or two minutes (simulation time step = 1-2 minutes) to limit the number of calculations.

Therefore, it is much more useful to create an alternate approximate model, lighter and easier to handle.

4.1.3 Approximated model

In the approximate model, all the layers of a specific material are summed up. The full battery model consists then on the elementary pattern described in section 4.1.1 but the thicknesses of each layer are the sum of all the layers of that material in the real model. These layers are much bigger than the ones from the real model and the mesh dimensions are bigger, respecting this way the mesh requirements. As a consequence, the number of finite elements and nodes is smaller and the resulting model is lighter. The geometry is built in a few seconds and the thermal analysis is performed quickly enough with a step time of 5 seconds. Therefore, the results obtained with the

approximate model follow the evolution of the temperature more accurately.

Consequently, the thermal analysis is performed on this model which offers more advantages than the very complex real one. The only corrections that have to be done to the original code are modifying the values of the thicknesses of the layers and deleting the VGEN commands, since the whole model is the elementary pattern itself.

The geometry of the approximate model is shown in the following figure:

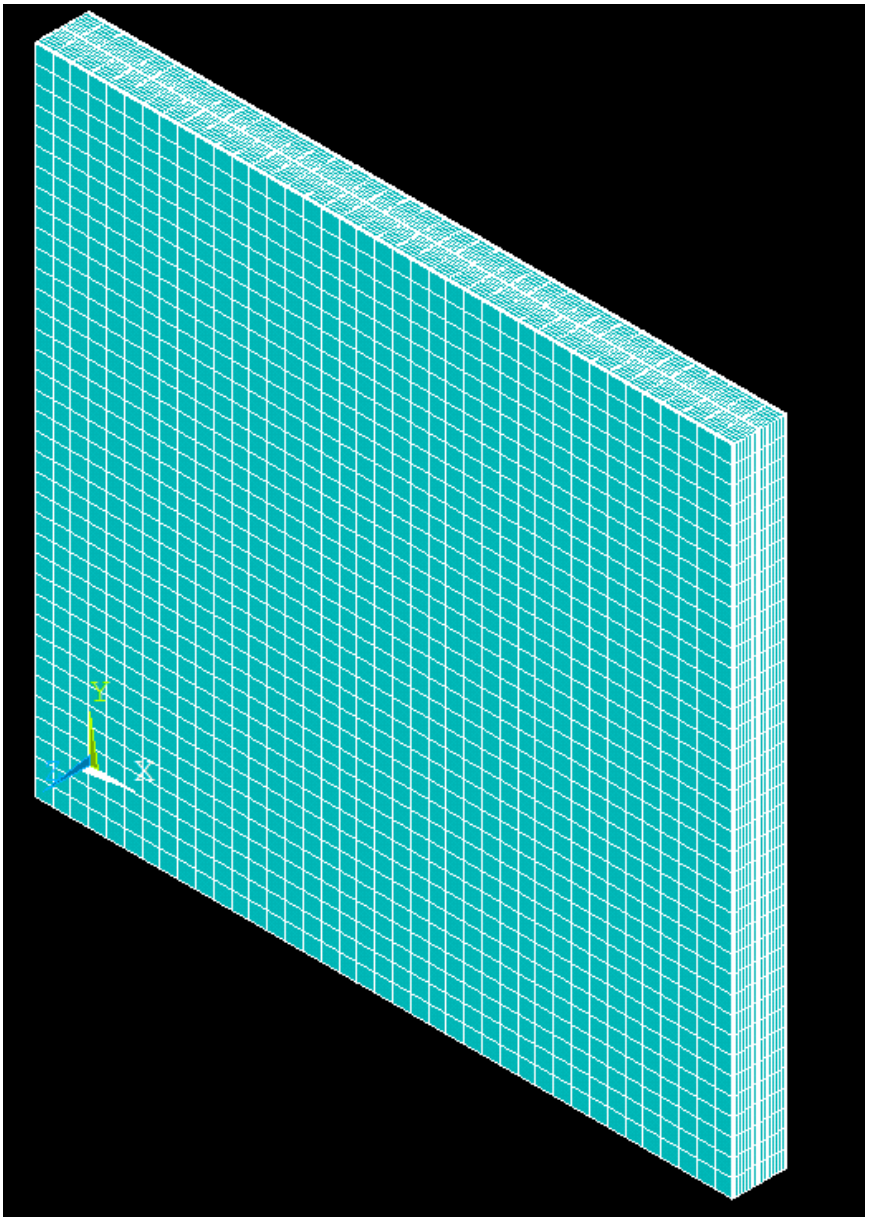


Figure 4 - Geometry of the approximate model.

The mesh of the model shown in the figure above is big enough for the finite elements to be clearly distinguishable. The different layers of material are also visible because the finite elements size varies in function of the layer thickness.

4.2 Thermal Analysis

4.2.1 Heat sources

It has been considered that the battery works in an environment without outside heat sources and at a room temperature of 298 K (around 25 Celsius degrees). Therefore, all the heat sources are within the battery and the only additional consideration that has to be made concerning the external conditions is the influence of the room temperature. In order to do so, the areas of the model in contact with the air are selected: only the separation areas between layers are completely out of the air influence and therefore are not selected. Once this is done, the influence of the room temperature is taken into account in the model by the means of the command SFA: it applies to the selected external body of the battery a convection load with a film coefficient of 15 W/m^2-K for the air.

Additionally, since the battery is considered to be in steady-state conditions at the beginning of the simulation, the temperature has to be equal to the room temperature at all the model nodes. This is done with the command TUNIF.

Inside the battery, the heat sources are the thermal losses explained above in section 2.1.3. The two different types of losses considered happen at the cathode and anode layers: the current circulating by the electrodes is the source of the losses by Joule effect and the thermodynamic losses are originated at the interface between the electrodes and the electrolyte. These losses are modelled by the means of a heat generation vector where the heat generation is defined as a function of the time. These vectors are applied as a body load to the electrode layers by the means of the command BFV.

The way the heat generation vectors are calculated and defined is explained next.

4.2.2 Solver stage

In this paper, it has been considered that the losses of the battery are around 5% of its nominal power, in this case 200W. Several simulations are launched with their amplitudes varying from 1 to 10W and with variable duration. The goal is to evaluate the behavior of the model for different inputs. Defining the load using a vector is very useful in this case since the duration and amplitude of the simulations heat generation can be changed very easily. The table 5 in section 5.1.2 sums up all the different simulations launched during the elaboration of the project.

In order to define the heat generation vector, the command *DIM is called: it defines the table hgen with n rows and one column. The number n of rows varies depending on the type of input required by the simulation given that at least two rows are required to completely define a single change in the input (upper and lower time boundaries). Two

additional rows are required to specify the initial and final state of the heat generation vector. Some examples are shown in Table 4, section 4.3.2.

The heat generation values are written in the one and only column of the table hgen. However these values alone are useless. To be meaningful, they need to be associated with some time points: every heat generation change occurs at a certain time and therefore it has to be associated with a certain time point. In order to do so, those points are specified into the zeroth column of hgen.

Additionally, it is mandatory to create an array defining the times profile (this can be done using the command *DIM too). When performing a heat transfer analysis with ANSYS, it is necessary to specify the key times at which the time-stepping strategy changes. In other words, the times when the vector hgen and therefore the body load applied to the model change have to be specified. This is done through the command TSRES. This new array is identical to the zeroth column of the table hgen above mentioned except for the first value since it is not necessary to specify the origin of times for this array (obviously, launching the simulation implies changing the body load of the model, even if it is zero). An example is provided below:

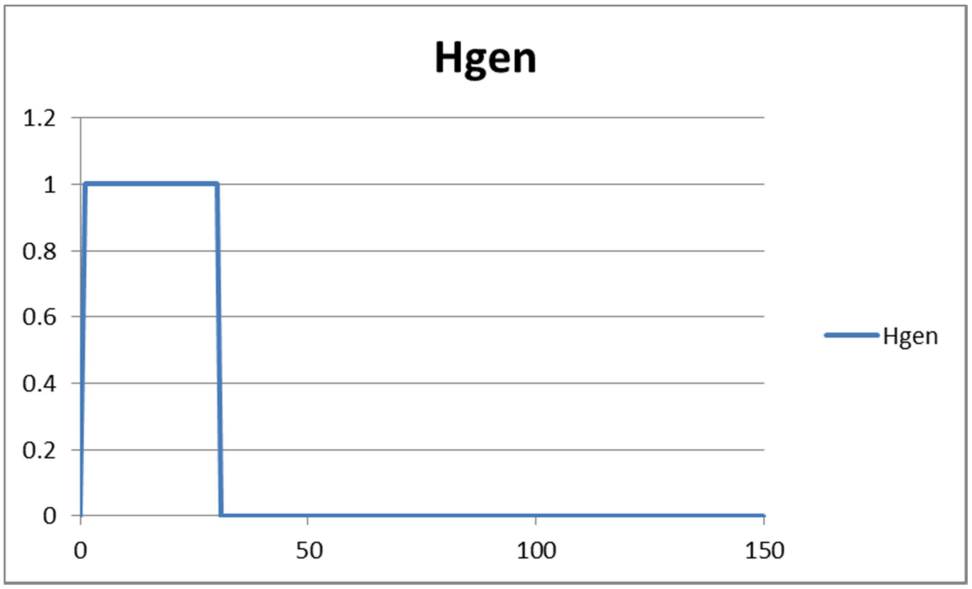


Figure 5 - Heat Generation load input: 1W during 30s

In this case, the table hgen has 4 rows, each one corresponding to the times 0, 1, 31 and 150 seconds (the final time of the simulation for the case of Figure 3).

As specified in section 3.2, Table 3, the heat generation in ANSYS is expressed as power function of volume, or W/m^3. Therefore, the values specified in hgen have to be expressed with this unit as well. However, the input applied to the model is the thermal power lost in the battery (in W). For that reason, it is necessary to convert the input to the actual values used in the table hgen by dividing it by the sum of the volumes of the anode and the cathode (since the heat is generated only at these layers). From this point, every heat generation mentioned in this paper is expressed in Watts but the real value

applied to the model is converted to W/m^3.

The system is pretty slow and the steady state is reached after a very long time: the final time for the simulations is 60000 seconds. As stated in 4.1.3, the step size used for the simulations is 5 seconds in order to make them quicker. As a consequence, some changes have to be made to the hgen table and the time array since the time values have to be multiples of 5.

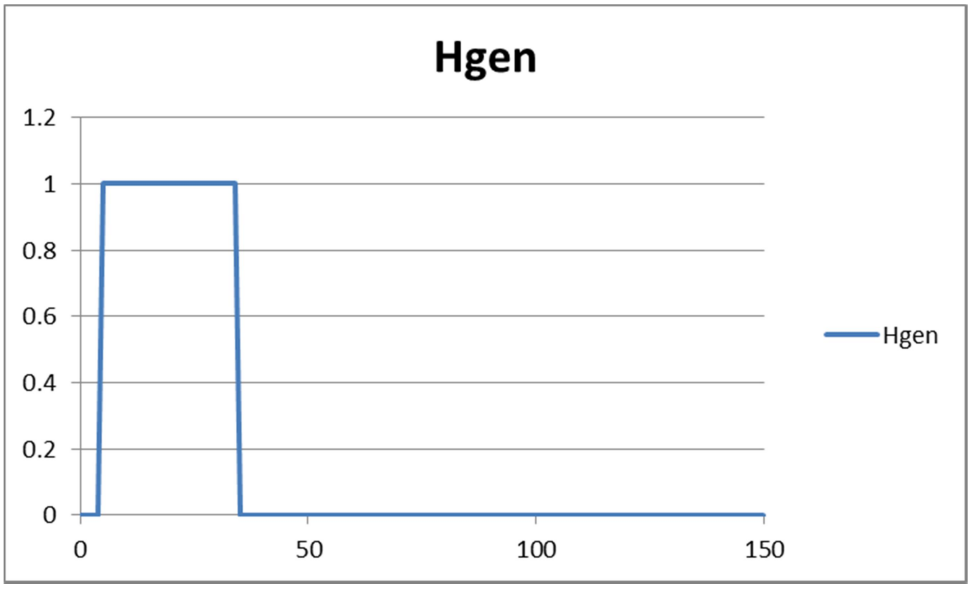


Figure 6 - Heat Generation load input: 1W during 30s with a time-step of 5s

In this case, the first step change occurs at t=5s. The specified hgen zeroth row values are then 0, 5, 35 and 150.

Finally, some parameters of the solver such as the type of variations of the field variables (step or ramp) and some transient analysis options are specified. Also, it is required through the command OUTRES that the solution written in the database is the one in the nodes for every sub step. The model is solved with the command SOLVE and once the solution is found, the solver stage ends. For the approximate model and under the time conditions explained above, the simulation is around 20 hours long.

4.3 Results

4.3.1 Division of the battery

ANSYS solves for every node of the model and each and every one of these solutions is accessible from the post-processing stage. However, the only considered solutions are the ones of the nodes located at the borders of the battery. This is because the other nodes are not accessible in a physical battery.

Even if the studied nodes are only a small fraction of the total number, it is not feasible

to consider all of them. So, in order to simplify the analysis of the solution, only 64 nodes are considered at the end: 32 at the frontal area of the battery and another 32 at the back. However, for simplicity's sake the results presented here concern only the front of the battery (the ones for the back are very similar). This division is shown in the figure below:

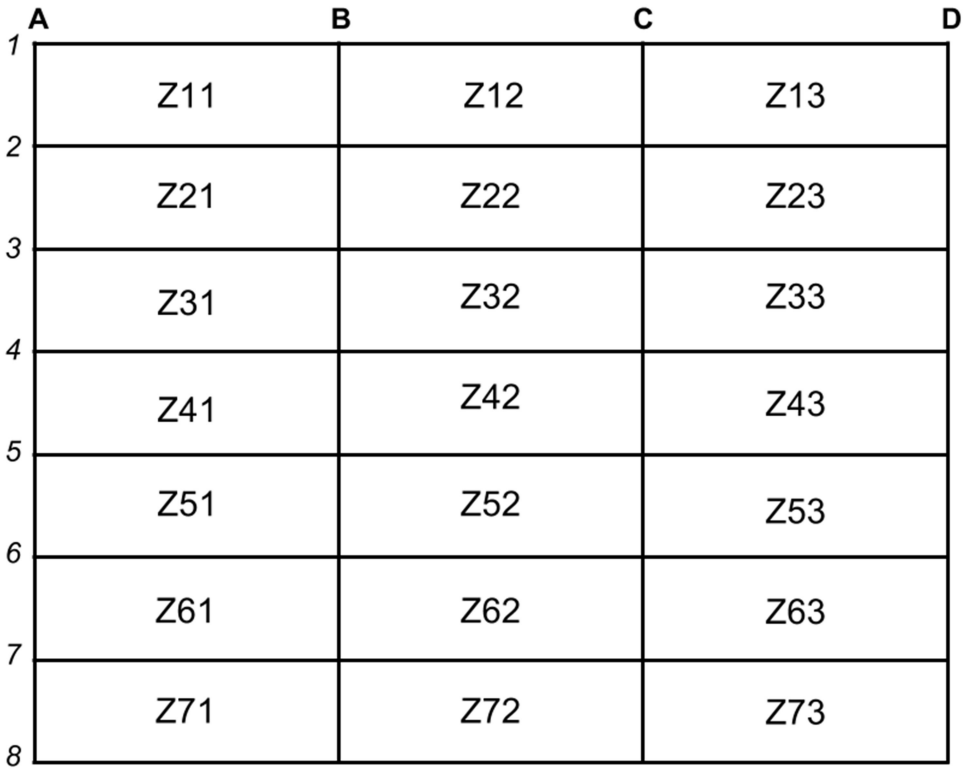


Figure 7 - Division of the battery.

This division is identical for both sides of the battery. As it can be seen, the different nodes divide the battery into 21 small areas. Considering that the temperature of each area is the mean of the temperatures of its 4 surrounding nodes, it is possible to see how the temperature is distributed in the battery and how it evolves over time, as it is shown in chapter 5.

Now that the interest nodes have been defined, it is time to extract the nodal solutions calculated and stored by ANSYS. In order to do so, it is necessary to enter the post-processing stage using the command /POST26. After creating these points using the nodes coordinates, the command PRVAR is used: it gives all the values of temperature as a function of time for all the desired nodes in a .txt file. The data is transferred to an Excel file in order to be treated. This process is done for both the front and the back of the battery in two different .xls files. For each simulation, a new Excel file is created to store the results.

4.3.2 Simulation for an impulse of 1W

The results of this particular simulation are very important since they are the required basis for applying the convolution. Because of this, the results of this simulation are stored in every Excel file, no matter the simulation stored in it. All these things are explained in depth in chapter 5.

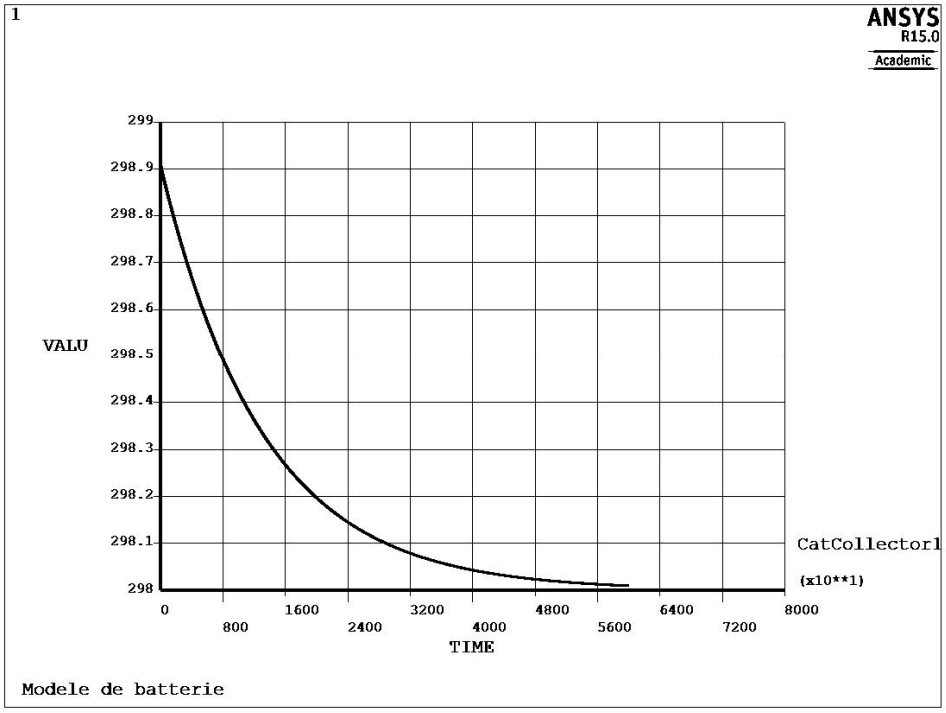


Figure 8 - Results for an input of 1W, 30s.

4.3.3 Other results

As follows the results of the simulations for different inputs.

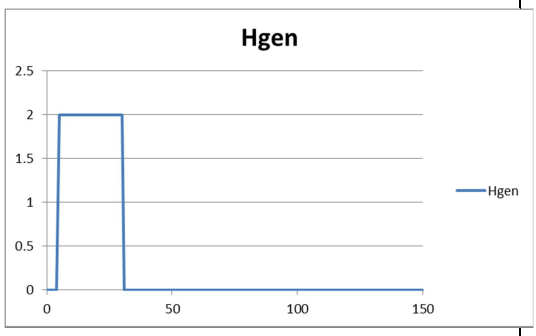
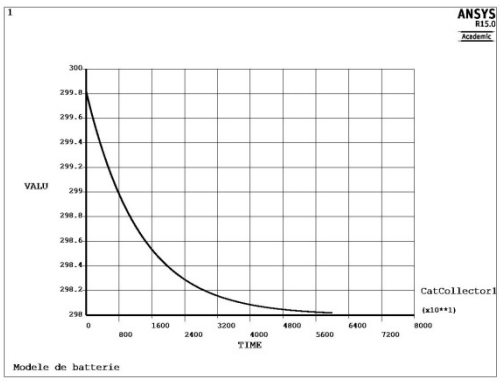
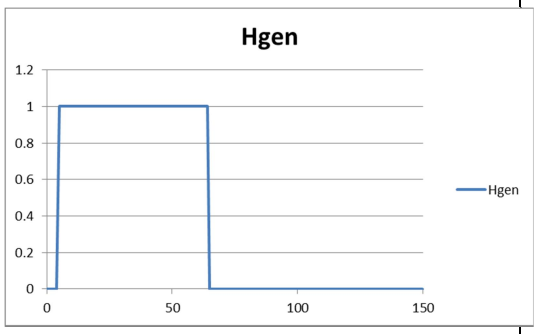
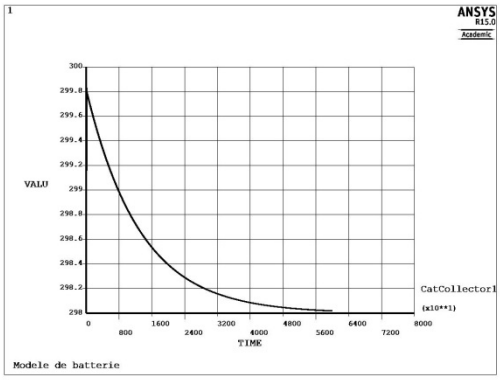
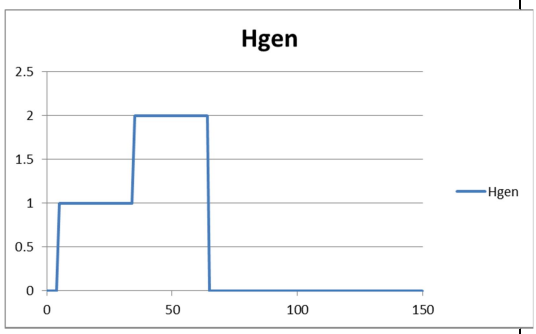
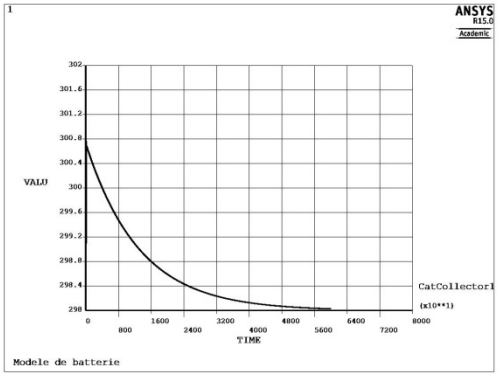
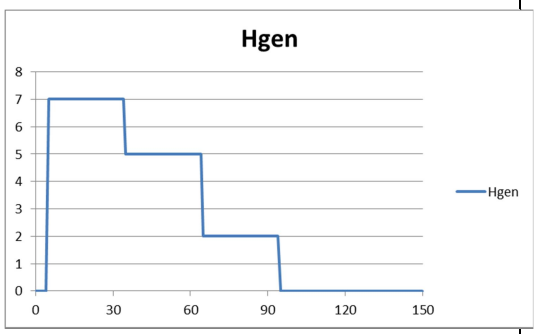
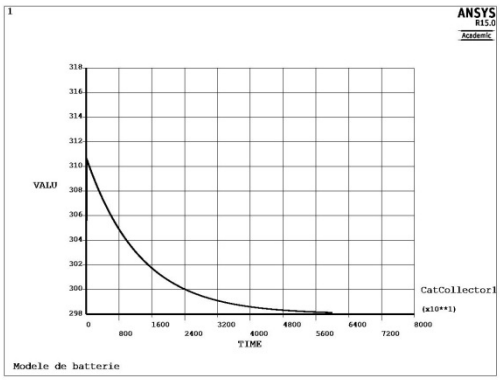
Impulsion	Input	Output
2W, 30s		
1W, 60s		
1W, 30s 2W, 30s		
7W, 30s 5W, 30s 2W, 30s		

Table 4 - Results from Ansys thermal analysis.

As it can be seen, the system behaves identically once the impulsion is over and the steady-state is reached at the same time. The only variations are the maximum temperature reached since it depends directly from the input applied to the model.

These results are used in chapter 5 as a comparative basis for the system response predictions, as it is explained there. These results can and have to be used as well to check the validity of our model by comparing them to real results from real battery tests.

5 The convolution theorem applied to the FE model.

This last chapter covers the convolution theorem and its application to the FE model previously created. In short, by the means of the convolution it is possible to predict the response of the model to a given input. First, the theorem and how it is applied are explained in detail alongside with the system of files and codes that allows an easier treatment of the data. Then, the results are presented and a conclusion about the validity of the application of the theorem to this case is drawn.

5.1 The convolution theorem

5.1.1 The theorem

The convolution product is an operation between functions with a huge importance in the Laplace transform theory. In particular, it is crucial for calculating inverse Laplace transforms. The convolution product of two functions is noted as $(f * g)(t)$ and defined by:

$$(f * g)(t) = \int_0^t f(u)g(t-u)du \tag{5.1}$$

Note that the asterisk denotes convolution in this context and not multiplication.

The most interesting property of the convolution is, referring to the Laplace transform:

$$\mathcal{L}[(f * g)(t)](s) = \mathcal{L}[f(t)](s)\mathcal{L}[g(t)](s) = F(s)G(s) \tag{5.2}$$

With F(s) and G(s) being the Laplace transform of f(t) and g(t) respectively. In other words, the Laplace transform of the convolution of two functions is the product of the Laplace transforms of those two functions.

There is a particularly interesting application concerning the Dirac delta function or the unit impulse function noted as $\delta(t)$. A property of this function states that, considering $f(t) = \delta(t)$, then its Laplace transform F(s) is equal to 1. In other words:

$$\mathcal{L}[\delta(t)](s) = 1 \tag{5.3}$$

Therefore, going back to equation 5.2:

$$\mathcal{L}[(f * \delta)(t)](s) = \mathcal{L}[f(t)](s) \cdot 1 = F(s) \tag{5.4}$$

And applying the inverse Laplace transform to both parts of the equation:

$$(f * \delta)(t) = f(t) \qquad \forall f(t) \tag{5.5}$$

Equations 5.2 and 5.5 are particularly important for this project as it is shown next.

5.1.2 The application

The battery can be modeled as a dynamic system with the following transfer function:

$$H(s) = \frac{Y(s)}{X(s)} \tag{5.6}$$

With:
$Y(s)$: nodal temperature (output of the system)
$X(s)$: heat generation (input of the system)
$H(s)$: transfer function of the system

And the equation of the system is then:

$$Y(s) = H(s)X(s) \tag{5.7}$$

Applying the inverse Laplace transform to (5.7) and with (5.2):

$$y(t) = h(t) * x(t) \tag{5.8}$$

The functions y(t) and x(t) are known, being respectively the results from the thermal analysis performed by ANSYS (temperature in Kelvin) and the input chosen for the simulation (heat generation in Watts). The goal here is to be able to predict y(t) for any x(t) and in order to do so, it is necessary to know h(t). It is necessary to note that both y(t) and x(t) are variation functions, not absolute ones. For x(t), this detail is not so important since the input on steady-state is 0 and therefore the absolute and the relative functions are the same. However, for y(t) the steady-state conditions are 298 degrees Kelvin which means that the obtained results will completely different depending on which function it is used.

Given that a time of 30 seconds is negligible compared to the 60000 seconds that every simulation last, it is possible to approximate an input of 1 Watt during 30 seconds as a Dirac delta function. Then considering the approximation above mentioned and the results of the simulation for this particular input, equation (5.8) becomes:

$$y_{1W,30s}(t) = h(t) * \delta(t) \tag{5.9}$$

And taking into consideration the equation (5.5):

$$y_{1W,30s}(t) = h(t) \tag{5.10}$$

Therefore, the transfer function of the system in the time domain is equal to the results of the simulation of the model for an input of 1W during 30s. It is now possible to predict the response y(t) of the system to any input x(t) calculating the convolution between this last function and the transfer function of the system in the time domain.

$$y(t) = y_{1W,30s}(t) * x(t) \tag{5.11}$$

Additional considerations have to be taken into account though. The convolution

product is a difficult operation, so it is done by the Matlab code "Convolution.m" presented in Appendix A. The input is written as a vector and since the time step of the simulation is 5 seconds, the different values of the vector are also separated by 5s. Following this logic, an impulse of 1W during 30s would be written as [1;1;1;1;1;1;0]. However, this is not a Dirac delta function since the impulse is present in more than one time point (in this case, in 6) and therefore the calculation performed by Matlab is not correct for this input.

In order to be able to apply the approximation above mentioned and consider the input as a Dirac delta function, it is required to perform a sampling of the results conducive to a time step change from 5 to 30 seconds. The sampling is done picking one value every six using a function for in Matlab. Once this is done, the values of the input vector are separated by 30 seconds and thus the same impulse of 1W during 30s is now written as [0;1;0] which corresponds to a Dirac impulsion. While this approximation may look a bit rough since the 12000 results that we originally got are reduced to 2000 by the sampling, it is a quite good one given that the evolution of the system temperature is very slow as it can be seen in section 4.3.2, table 4.

During the realization of this project, a lot of simulations were launched. They are presented next with their respective input vector used for the code "Convolution.m".

1W, 30s	[1;0]	5W, 60s	[5;5;0]
2W, 30s	[2;0]	8W, 60s	[8;8;0]
5W, 30s	[5;0]	10W, 60s	[10;10;0]
8W, 30s	[8;0]	1W, 2W, 30s	[1;2;0]
10W, 30s	[10;0]	5W, 8W, 30s	[5;8;0]
1W, 60s	[1;1;0]	7W, 5W, 2W, 30s	[7;5;2;0]
2W, 60s	[2;2;0]	7W, 0W, 5W, 0W, 2W	[7;0;5;0;2;0]

Table 5 - List of simulations launched.

Additionally, the different inputs of some simulations are presented next, alongside with its sampled equivalent.

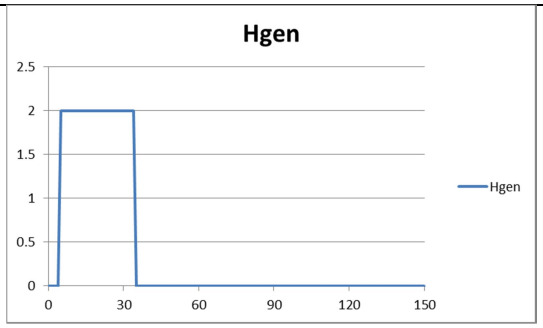
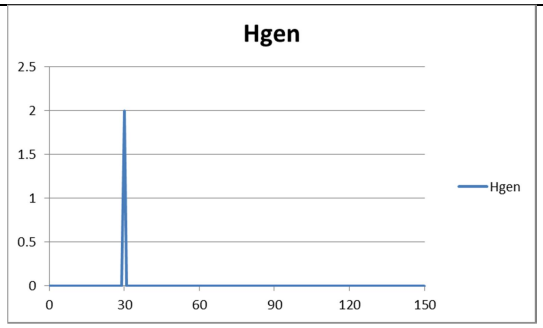
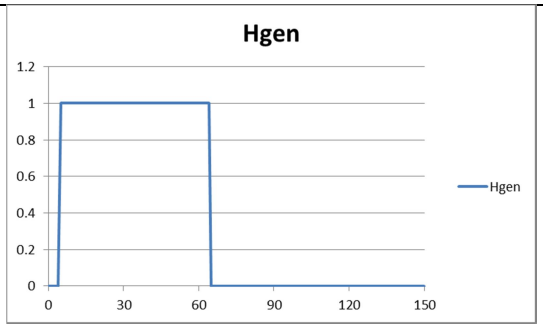
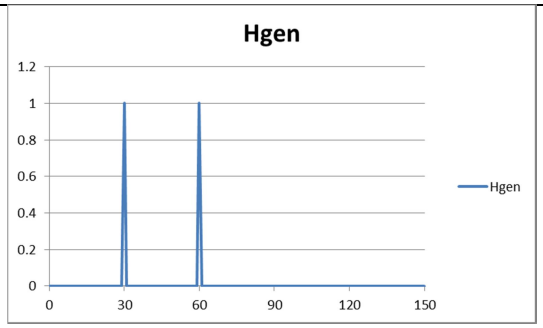
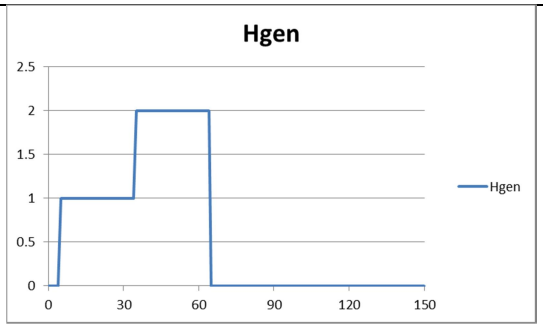
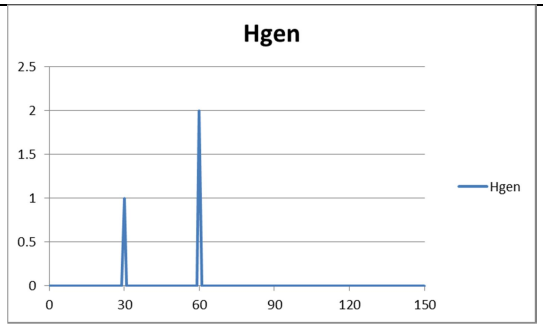
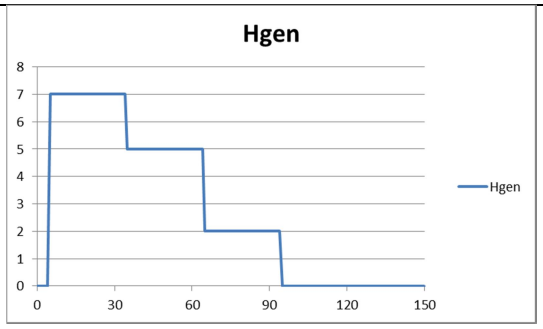
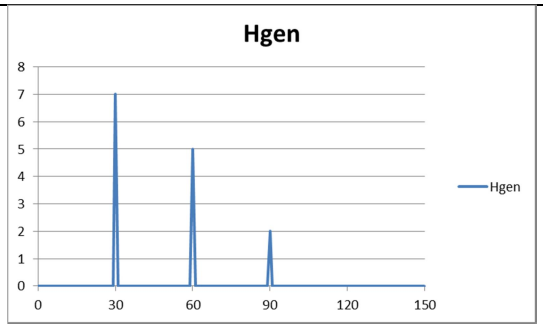
Impulsion	Raw input	Sampled input
2W, 30s [2;0]		
1W, 60s [1;1;0]		
1W, 30s 2W, 30s [1;2;0]		
7W, 30s 5W, 30s 2W, 30s [7;5;2;0]		

Table 6 - Raw and sampled inputs.

As it can be seen, the sampling takes one value every 30 seconds, or one value out of six. The selected one is the last value of each time interval of 30s.

The code “Convolution.m” samples the results, calculates the convolution and stores the results in an Excel file all at once.

5.1.3 The system Excel-Matlab

The process of sampling the thermal analysis results, calculating the convolution,

storing the resulting data and treating it to get the results has to be done for the 64 nodes considered of the model. It is a tiresome and nearly infeasible task if it is done node-by-node that may take as long as the thermal analysis itself. In order to prevent this and to optimize this operation, a nearly automatized system has been designed. In this system, the operator only has to adjust the input for the code "Convolution.m" and to specify which type of result he would like to obtain from the other two codes (explained in the next section). The cornerstone of this system is the Excel file, since it is the "storage room" of all the data used by the Matlab scripts and operates also as a conductive bridge between the different codes.

Two systems Excel-Matlab have been considered for each simulation: one for the front, and another for the back of the battery. The whole system is contained within a folder, where the demanded results are also located as can be seen below.

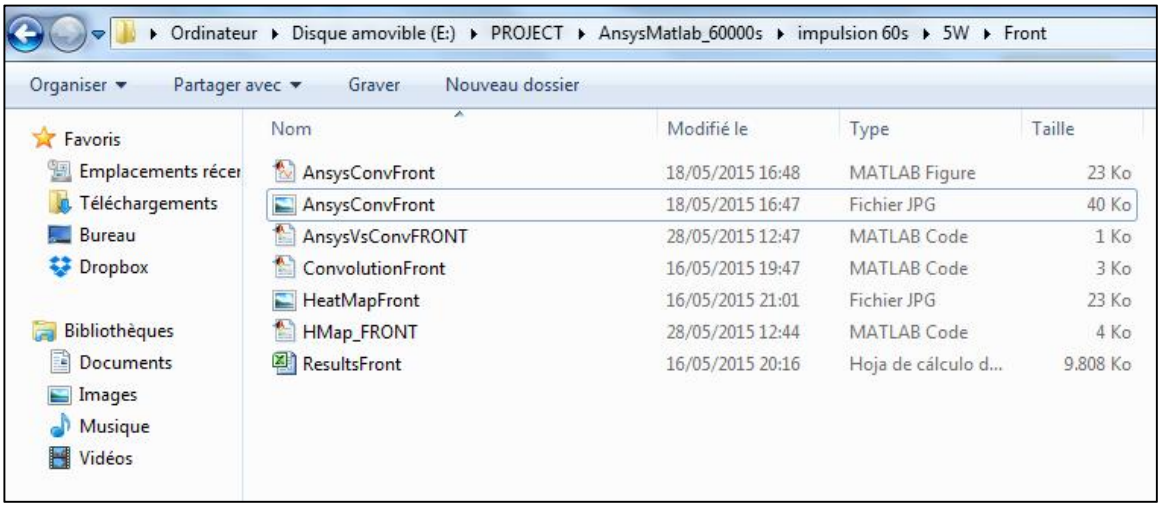


Figure 9 - Folder with the system Excel-Matlab.

As it was said before, the Excel file contains the results of the thermal analysis on ANSYS for every considered node of the battery and it is divided in four sheets:

- The first one contains the results of the simulation for an input of 1W, 30 seconds. In other words, it contains the system transfer function in the real time domain. This sheet does not depend on the studied simulation.
- The second one contains the results of the studied simulation: the response of the system to the considered input. Its values are inputs to the "Convolution.m" code: they are sampled and then returned to the fourth sheet of the file.
- The third sheet contains the temperature variation for the 1W, 30 seconds input. It does not depend on the studied simulation either. The sole goal of this sheet is to simplify a bit the code "Convolution.m". Its values are also inputs to that code.
- The four and final sheet contains the results from the studied simulation and from the convolution calculation, both sampled and written by "Convolution.m" (thus there are less values in this sheet than in the other three). It also calculates the relative error between the two methods value-to-value.

5.2 Results

5.2.1 Heat Map

The data required to obtain the results is written by the Matlab codes from the Excel. One of the results that have been extracted from the analysis is a heat map. Following the distribution specified in section 4.3.1, figure 7, the heat map shows how the temperature distributes on the battery. It can show it for a given instant, specified by the programmer, or for the whole simulation. The treatment of the results and the drawing of the heat map are tasks done by the code "HMap.m", detailed in Appendix B. An example is shown below.

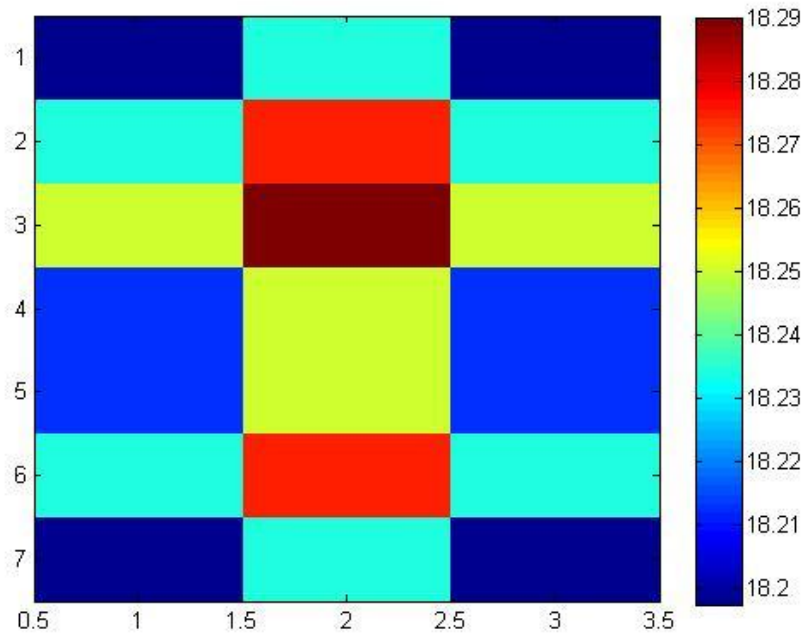


Figure 10 - Heat Map for an input of 10W during 60s at t=60s

The different zones are easily identifiable. It can be seen that the differences of temperature between zones are little and thus, the temperature is distributed in the battery in a quite homogenous way. Additionally, this distribution pattern does not vary depending on the impulsion as it is shown below.

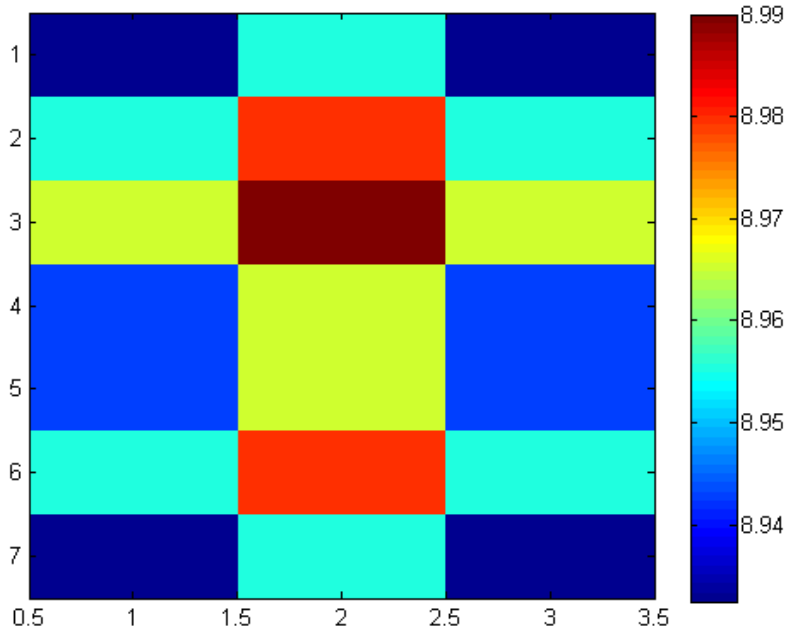


Figure 11 - Heat Map for an input of 10W during 30s at t=30s.

It can be seen that while the temperature value and the difference between the coldest and the hottest zones varies, the distribution pattern stays the same.

5.2.2 Comparison of the results

Comparing the results obtained with Ansys with those obtained with the convolution is a very important task since the validity of our model and of our predictions relays on it. The comparison is done both analytically (Excel) and graphically (Matlab).

The Excel file stores in its fourth sheet the sampled results from the model thermal analysis and from the convolution calculation, so the comparison is pretty straight-forward. The variables compared here are the nodal temperatures. Some examples are shown below.

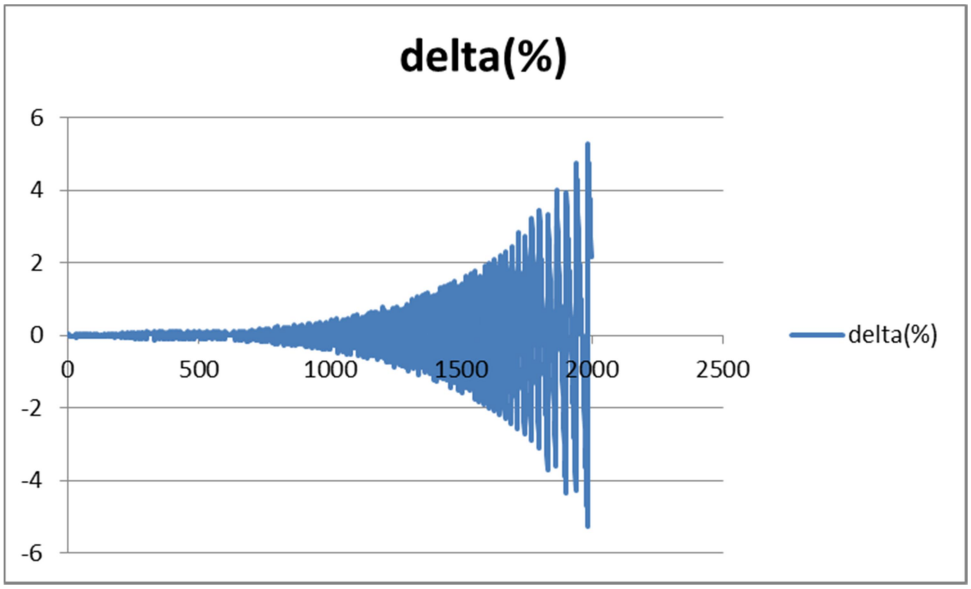


Figure 12 – Relative error between ANSYS and the convolution calculation for the simulation of 10W during 60s at node D4.

The relative error stays below 5% at all times. The error grows bigger and bigger as the simulation lasts longer and longer. This is due to the fact that the absolute error stays the same at all times while the temperature always goes down once the input has stopped. As a consequence, the percentage error is bigger in the late stages of the simulation than in the early ones. The (relative) constant behavior of the absolute error can be seen in the following graph.

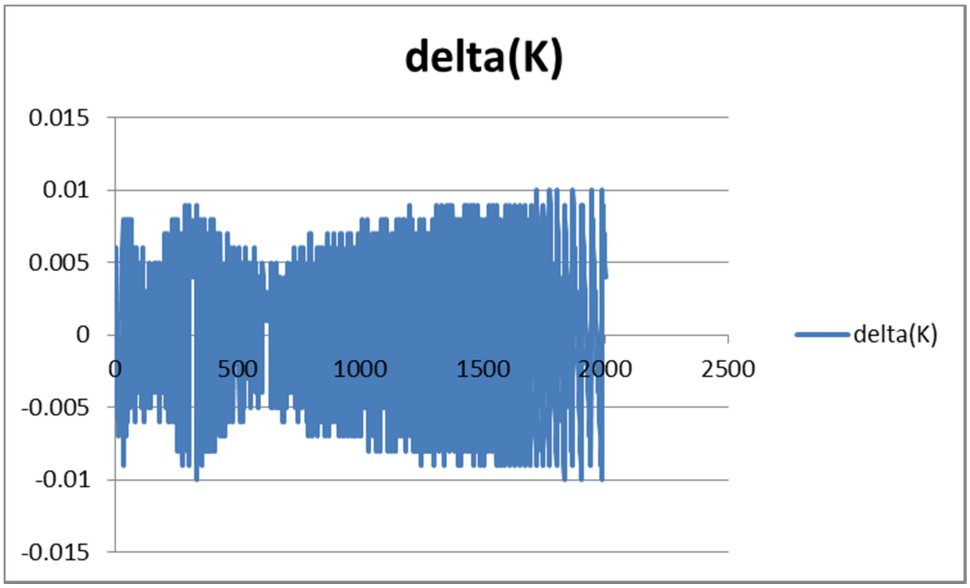


Figure 13 – Absolute error between ANSYS and the convolution calculation for the simulation of 10W during 60s at node D4.

As it is shown above, the absolute error is always between -0.01 and 0.01 Kelvin degrees.

Additionally, the error between the ANSYS and the convolution calculations does not

depend on the input:

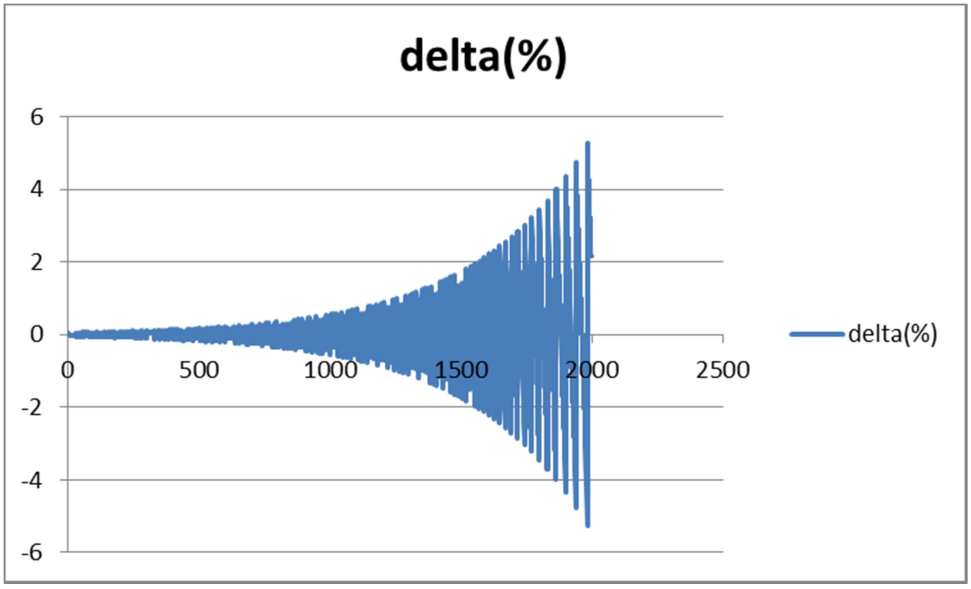


Figure 14 - Relative error between ANSYS and the convolution calculation for the simulation of 10W during 30s at node D4.

It can be seen that the relative error for the same node with a different input has the same shape as before.

The two results can also be compared graphically using a Matlab script. This code, called "AnsysVsConv.m", is detailed at Appendix C. In opposition to the analytical method, the code compares the zonal temperatures and not the nodal ones. As "HMap.m", it reads the required data from the fourth sheet of the Excel file, it calculates the temperatures and finally it creates the graph.

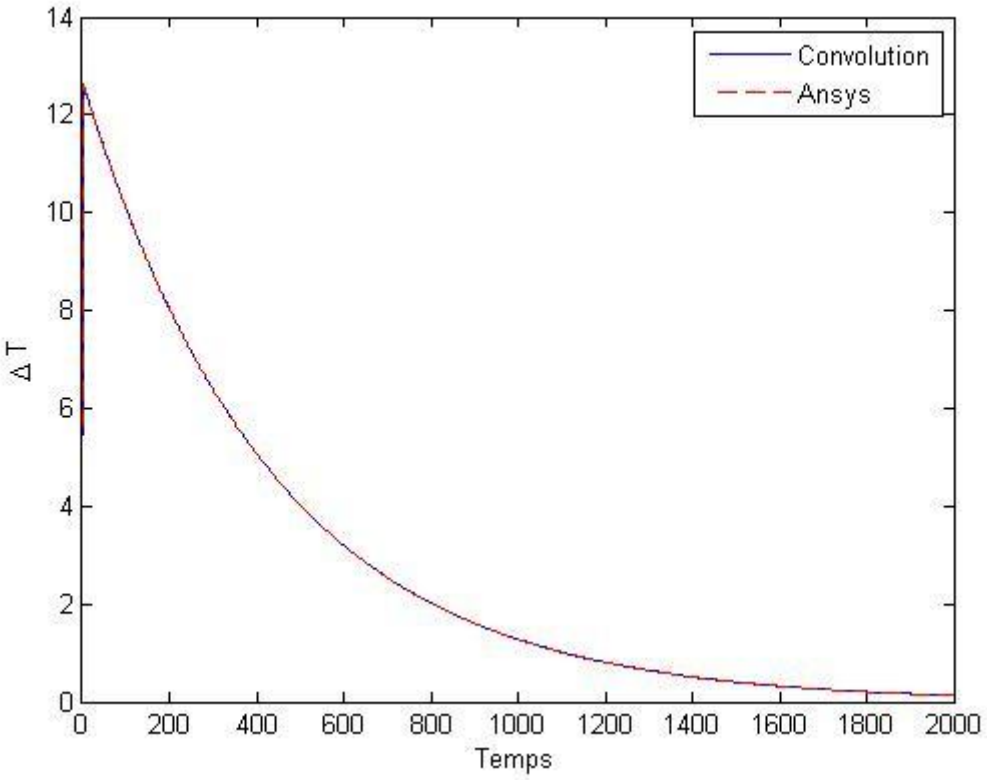


Figure 15 - Comparison of the results obtained with Ansys and with the convolution for the simulation of 7W, 30s, 5W, 30s and 2W, 30s for the zone Z32.

The two different graphs are indistinguishable: the error is very little. Because of the sampling, the most problematic zone of the graph in terms of error should be the one where the variations of the temperature are more abrupt, since the sampling carries a sensibility loss. In other words, the error may be bigger in the early stages of the simulation and particularly in the peak of temperature.

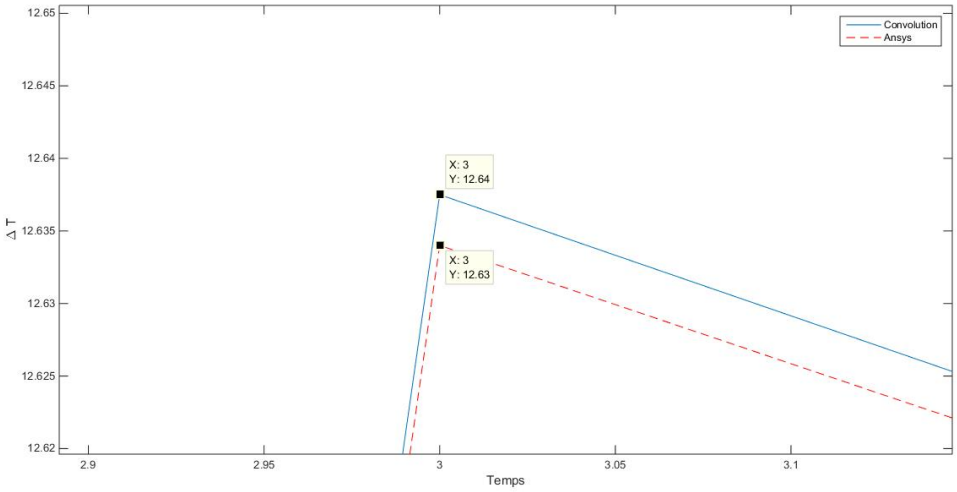


Figure 16 - Comparison of the results obtained with Ansys and with the convolution for the simulation of 7W, 30s, 5W, 30s and 2W, 30s for the zone Z32 at t=3.

As shown above, the error at the peak at t=3 (that is, t=90 seconds in the real world) is almost negligible.

6 Conclusions

At this work, a thermal FE model of a Li-ion battery has been built. The model is not an exact replicate of the real battery but an approximation. This decision was made in order to simplify the problem and to reduce the simulation time. After defining the different thermal loads affecting the model and solving the problem, some particularly relevant nodal results have been selected in order to be able to know the thermal behavior of the battery while keeping the problem as simple as possible. The model has been tested for a huge variety of different inputs with one of them being particularly important: the response of the model for an input of 1W during 30 seconds. All the results have been stored in Excel files for further use.

After this, the convolution theorem has been introduced and applied to the model. Using the results for the input of 1W, 30s it has been possible to obtain the system transfer function in the real time domain. With this parameter known, it is possible to calculate and predict the response of the model (temperature evolution on targeted points) for any considered input (heat generation rate). This method has been applied for all the inputs simulated on Ansys in the first part and then compared with the thermal analysis results. It has shown that both results are almost identical and therefore that it is possible to predict the behavior of the system using the convolution theorem.

Once the FE and forecast models are built and theoretically confirmed as correct, it is time to use them in real applications. Further work should compare the thermal behavior of a real battery for a given input to the model one. Once the thermal FE model is verified, it should be possible to predict the behavior of a real energy storage using the convolution theorem.

There is already at DESL-EPFL an ongoing activitiy associated to the experimental validation of the convolution theorem for the temperature prediction. The first results are very interesting and promising for a validation of the model described in this work.

BIBLIOGRAPHY

[1]: Battery University. http://batteryuniversity.com/learn

[2]: Dell, R. M. and Rand, D. A, "Understanding Batteries", Royal Society of Chemistry, 2001.

[3]: Hémery, C. V., "Etudes des phénomènes thermiques dans les batteries Li-ion", Université de Grenoble, 2013.

[4]: Accuratus website. http://accuratus.com/alumox.html

[5]: Fluorotherm website. http://www.fluorotherm.com/technical-information/materials-overview/pvdf-properties/

[6]: Chen, M. et al., "A Thermal Runaway Simulation on a Lithium Titanate Battery and the Battery Module", Energies, vol. 8, issue 1, pp. 490-500, 2015.

[7]: Kim, Y. and Kim, D., "Synthesis of High Density Nickel Cobalt Aluminum Hydroxide by Continuous Coprecipitation Method", American Chemical Society, pp. 586-589, 2012.

[8]: Maleki, H. et al., "Thermal Properties of Lithium-Ion Battery and Components", Journal of Electrochemical Society, vol. 146, issue 3, pp. 947-954, 1999.

[9]: Bhakivatti, S. S., "Finite Element Analysis", New Delhi: New Age International, 2005.

[10]: "Thermal Analysis Guide", ANSYS, Inc, 2009.

[11]: "Theory Reference for the Mechanical APDL and Mechanical Applications", ANSYS, Inc, 2009.

[12]: García, A., García, F., López, A., Rodríguez, G., de la Villa, A., "Ecuaciones Diferenciales Ordinarias. Teoría y Problemas", Clagsa, 2006.

[13]: Xiao, Y., Torregrossa, D., Paolone, M., "Surface temperatura estimation of li-ion battery via thermal impulse response technique", APEC, pp. 1089-1095, 2015.

APPENDIX A

"Convolution.m"

The goal of this code is to apply the convolution theorem in order to obtain the response of the system to a certain impulse x (in this case, 1W during 60 seconds). The code reads the results of the thermal analysis in ANSYS for the case of 1W during 30 seconds stored in an Excel file, samples it changing the time variation from 1s to 30s and then it operates the convolution. The results are stored in the same Excel file. Similarly, the results of the thermal analysis for the impulse being studied are also read, sampled and then re-stored in the same Excel document in order to perform a numerical comparison between both results.

NOTE: As stated in 4.3.1, the results for the front and the back of the battery are studied for each of the simulations in different Excel files. Since these files are directly related to the Matlab codes, there exist two different Convolution.m codes: one for the front and another for the back. The code presented next is used to study the results of the frontal nodes.

```
%Convolution a Excel
clear all
clc


x=[1;1;0];     %IMPULSION (Á MODIFIER)



lettres=char(65:90);

%On a 26 lettres mais 32 points. Il faut utiliser deux
fonctions 'for'.

for i=2:length(lettres); %De B a Z


h=xlsread('ResultsFront.xlsx',3,sprintf('%s4:%s12003',lett
res(i),lettres(i)));

yreal=xlsread('ResultsFront.xlsx',2,sprintf('%s4:%s12003',
lettres(i),lettres(i)));
```

```
    for j=1:12000/6;   %Cambio de base

        hnew(j)=h(j*6);

    end

    Temp=conv(x,hnew);
    yexcel=Temp';

    for j=1:12000/6;

        yrealnew(j)=yreal(j*6);

    end

    yrexcel=(yrealnew-298)';

    if (3*i-4<=26)


xlswrite('ResultsFront.xlsx',yexcel,4,sprintf('%s3',lettre
s(3*i-5)));

xlswrite('ResultsFront.xlsx',yrexcel,4,sprintf('%s3',lettr
es(3*i-4)));

    elseif (3*i-30<=26)


xlswrite('ResultsFront.xlsx',yexcel,4,sprintf('%s%s3',lett
res(1),lettres(3*i-31)));

xlswrite('ResultsFront.xlsx',yrexcel,4,sprintf('%s%s3',let
tres(1),lettres(3*i-30)));

    elseif (3*i-56<=26)

        if (3*i-57==0)

xlswrite('ResultsFront.xlsx',yexcel,4,sprintf('%s%s3',lett
res(1),lettres(3*i-31))); %Case of AZ
        else

xlswrite('ResultsFront.xlsx',yexcel,4,sprintf('%s%s3',lett
res(2),lettres(3*i-57))); % Case of B-Else
        end


xlswrite('ResultsFront.xlsx',yrexcel,4,sprintf('%s%s3',let
```

```
tres(2),lettres(3*i-56)));

    end
end

for i=1:7;  %De AA a AG


h=xlsread('ResultsFront.xlsx',3,sprintf('%s%s4:%s%s12003',
lettres(1),lettres(i),lettres(1),lettres(i)));

yreal=xlsread('ResultsFront.xlsx',2,sprintf('%s%s4:%s%s120
03',lettres(1),lettres(i),lettres(1),lettres(i)));

    for j=1:12000/6;   %Cambio de base

        hnew(j)=h(j*6);

    end

    Temp=conv(x,hnew);
    yexcel=Temp';

    for j=1:12000/6;

        yrealnew(j)=yreal(j*6);

    end

    yrexcel=(yrealnew-298)';

    if (3*i+22<=26)  %+21 porque hemos restado 26 de i - 5
de formula original.


xlswrite('ResultsFront.xlsx',yexcel,4,sprintf('%s%s3',lett
res(2),lettres(3*i+21)));

xlswrite('ResultsFront.xlsx',yrexcel,4,sprintf('%s%s3',let
tres(2),lettres(3*i+22)));

    else

xlswrite('ResultsFront.xlsx',yexcel,4,sprintf('%s%s3',lett
res(3),lettres(3*i-5)));

xlswrite('ResultsFront.xlsx',yrexcel,4,sprintf('%s%s3',let
tres(3),lettres(3*i-4)));

    end
```

```
end
```

APPENDIX B

"HMap.m"

In order to run this code, it is mandatory to have previously run "Convolution.m". The goal of HMap.m is, as its name says, to build a heat map of the battery using the nodal temperatures obtained from the Ansys thermal analysis or from the application of the convolution theorem. It can be done for a given instant chosen by the user or for the complete time lapse: in this last case, the output of the program is a video in .avi format. The different nodes and zones used for building the heat map are shown in section 4.3.1, figure 7.

As stated in Appendix A, there exist two HMap.m codes for each simulation. The example presented below builds the heat map for the front side of the battery.

```
%HEAT MAP


clear all
clc

%IF YOU WANT THE HEATMAP JUST FOR THE MAX. TEMP INSTANT,
MODIFY i
i=1;

%IF YOU WANT AN ANIMATED HEATMAP, DELETE THE I AND USE THE
FOR.


%DATA
a1=xlsread('ResultsFront.xlsx',4,'A4:A2003');
a2=xlsread('ResultsFront.xlsx',4,'D4:D2003');
a3=xlsread('ResultsFront.xlsx',4,'G4:G2003');
a4=xlsread('ResultsFront.xlsx',4,'J4:J2003');
a5=xlsread('ResultsFront.xlsx',4,'M4:M2003');
a6=xlsread('ResultsFront.xlsx',4,'P4:P2003');
a7=xlsread('ResultsFront.xlsx',4,'S4:S2003');
a8=xlsread('ResultsFront.xlsx',4,'V4:V2003');
b1=xlsread('ResultsFront.xlsx',4,'Y4:Y2003');
b2=xlsread('ResultsFront.xlsx',4,'AB4:AB2003');
b3=xlsread('ResultsFront.xlsx',4,'AE4:AE2003');
b4=xlsread('ResultsFront.xlsx',4,'AH4:AH2003');
b5=xlsread('ResultsFront.xlsx',4,'AK4:AK2003');
```

```
b6=xlsread('ResultsFront.xlsx',4,'AN4:AN2003');
b7=xlsread('ResultsFront.xlsx',4,'AQ4:AQ2003');
b8=xlsread('ResultsFront.xlsx',4,'AT4:AT2003');
c1=xlsread('ResultsFront.xlsx',4,'AW4:AW2003');
c2=xlsread('ResultsFront.xlsx',4,'AZ4:AZ2003');
c3=xlsread('ResultsFront.xlsx',4,'BC4:BC2003');
c4=xlsread('ResultsFront.xlsx',4,'BF4:BF2003');
c5=xlsread('ResultsFront.xlsx',4,'BI4:BI2003');
c6=xlsread('ResultsFront.xlsx',4,'BL4:BL2003');
c7=xlsread('ResultsFront.xlsx',4,'BO4:BO2003');
c8=xlsread('ResultsFront.xlsx',4,'BR4:BR2003');
d1=xlsread('ResultsFront.xlsx',4,'BU4:BU2003');
d2=xlsread('ResultsFront.xlsx',4,'BX4:BX2003');
d3=xlsread('ResultsFront.xlsx',4,'CA4:CA2003');
d4=xlsread('ResultsFront.xlsx',4,'CD4:CD2003');
d5=xlsread('ResultsFront.xlsx',4,'CG4:CG2003');
d6=xlsread('ResultsFront.xlsx',4,'CJ4:CJ2003');
d7=xlsread('ResultsFront.xlsx',4,'CM4:CM2003');
d8=xlsread('ResultsFront.xlsx',4,'CP4:CP2003');


%Defining the model
%for i=1:1999;
    %On definit les carres.
    %Chaque carre est la moyenne de ces 4 sommets.

    z11(i)=(a1(i)+a2(i)+b1(i)+b2(i))/4;
    z12(i)=(b1(i)+b2(i)+c1(i)+c2(i))/4;
    z13(i)=(c1(i)+c2(i)+d1(i)+d2(i))/4;

    z21(i)=(a2(i)+a3(i)+b2(i)+b3(i))/4;
    z22(i)=(b2(i)+b3(i)+c2(i)+c3(i))/4;
    z23(i)=(c2(i)+c3(i)+d2(i)+d3(i))/4;

    z31(i)=(a3(i)+a4(i)+b3(i)+b4(i))/4;
    z32(i)=(b3(i)+b4(i)+c3(i)+c4(i))/4;
    z33(i)=(c3(i)+c4(i)+d3(i)+d4(i))/4;

    z41(i)=(a4(i)+a5(i)+b4(i)+b5(i))/4;
    z42(i)=(b4(i)+b5(i)+c4(i)+c5(i))/4;
    z43(i)=(c4(i)+c5(i)+d4(i)+d5(i))/4;

    z51(i)=(a5(i)+a6(i)+b5(i)+b6(i))/4;
    z52(i)=(b5(i)+b6(i)+c5(i)+c6(i))/4;
    z53(i)=(c5(i)+c6(i)+d5(i)+d6(i))/4;

    z61(i)=(a6(i)+a7(i)+b6(i)+b7(i))/4;
    z62(i)=(b6(i)+b7(i)+c6(i)+c7(i))/4;
    z63(i)=(c6(i)+c7(i)+d6(i)+d7(i))/4;
```

```
    z71(i)=(a7(i)+a8(i)+b7(i)+b8(i))/4;
    z72(i)=(b7(i)+b8(i)+c7(i)+c8(i))/4;
    z73(i)=(c7(i)+c8(i)+d7(i)+d8(i))/4;


    Data(:,:,i)=[z11(i) z12(i) z13(i);
        z21(i) z22(i) z23(i);
        z31(i) z32(i) z33(i);
        z41(i) z42(i) z43(i);
        z51(i) z52(i) z53(i);
        z61(i) z62(i) z63(i);
        z71(i) z72(i) z73(i)];

%end


%Building the battery

%A single heat map

%The temp. peak is i=1 for 30s and i=2 for 60s and 2x30s.

A=Data(:,:,i);
colormap('default');
imagesc(A);
colorbar;

%Heat Map movie

% F(1999)=struct('cdata',[],'colormap',[]);     %We create
an array.
%          %cdata          is          a          field:
http://ch.mathworks.com/help/matlab/ref/struct.html
%
% for i=1:1999
%     A=Data(:,:,i);
%     colormap('default');
%     imagesc(A);
%     colorbar;
%     %caxis([0.1 1.9])    %Fix the axes.
%     F(i)=getframe(gcf);
% end
%
%
% movie2avi(F, 'HeatMap_front_notfixed');
```

APPENDIX C

"AnsysVsConv.m"

Before running this code, it is also necessary to previously run Convolution.m. The goal of this last code is to compare graphically the results obtained from the ANSYS thermal analysis and from the application of the convolution theorem. It compares the temperatures of one of the zones defined in section 4.3.1, figure 7, previously chosen by the programmer. After calculating the required zonal temperatures from the nodal ones for both cases, it creates a comparative graph.

As stated in Appendix A, there exist two AnsysVsConv.m codes for each simulation. The example presented below builds the graph for the front side of the battery.

```
%Ansys vs convolution
%EXAMPLE FAIT POUR Z32.

%Convolution (copier de HMap-Front.m)
b4=xlsread('ResultsFront.xlsx',4,'AH4:AH2003');
b3=xlsread('ResultsFront.xlsx',4,'AE3:AE2003');
c4=xlsread('ResultsFront.xlsx',4,'BF3:BF2003');
c3=xlsread('ResultsFront.xlsx',4,'BC3:BC2003');

%Ansys (une lettre "en haut")
x1=xlsread('ResultsFront.xlsx',4,'AI3:AI2003');
x2=xlsread('ResultsFront.xlsx',4,'AF3:AF2003');
x3=xlsread('ResultsFront.xlsx',4,'BG3:BG2003');
x4=xlsread('ResultsFront.xlsx',4,'BD3:BD2003');

for i=1:1999

    z32(i)=(b4(i)+b3(i)+c4(i)+c3(i))/4;
    x32(i)=(x1(i)+x2(i)+x3(i)+x4(i))/4;

end

%RESULTATS CONVOLUTION
plot(z32);
hold on;
plot(x32,'r--');
legend('Convolution','Ansys');
xlabel('Temps');
ylabel('\Delta T');
```