



**Universidad
Zaragoza**

TRABAJO FIN DE GRADO

Estudio termodinámico de una pila de combustible de membrana polimérica de alta temperatura (HTPEMFC) mediante simulación numérica en códigos de fuentes libres (OpenFOAM).

Thermodynamic study of a high temperature polymeric membrane fuel cell (HTPEMFC) by numerical simulation in open source code (OpenFOAM).

Autor

Eduardo López Gimeno

Directores

Radu Mustata Oroviceanu

Raul Losantos Viñuales

Titulación del autor: Ingeniería Mecánica

ESCUELA DE INGENIERÍA Y ARQUITECTURA

Año 2020



DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe entregarse en la Secretaría de la EINA, dentro del plazo de depósito del TFG/TFM para su evaluación).

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

D./D^a. Eduardo López Gimemo, en
aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de
septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el
Reglamento de los TFG y TFM de la Universidad de Zaragoza,
Declaro que el presente Trabajo de Fin de (Grado/Máster)
Grado (Título del Trabajo)

Estudio termodinámico de una pila de combustible de membrana polimérica de alta
temperatura (HTPEMFC) mediante simulación numérica en códigos de fuentes libres
(OpenFOAM).

es de mi autoría y es original, no habiéndose utilizado fuente sin ser
citada debidamente.

Zaragoza, 19 de noviembre de 2020

Fdo: Eduardo López Gimemo

RESUMEN

Estudio termodinámico de una pila de combustible de membrana polimérica de alta temperatura (HTPEMFC) mediante simulación numérica en códigos de fuentes libres (OpenFOAM).

En este trabajo se va a añadir al código existente un módulo de cálculo previo que realice un cálculo aproximado de la temperatura a la que se encuentra la pila, con el objetivo de reducir los tiempos de computación de la simulación. También se va a añadir una condición de contorno que simule la disipación de calor mediante convección natural.

Primero se realizará una introducción a la materia en el capítulo 1, donde se explicará que son las pilas de combustible, también se explicará que es el programa OpenFOAM y como trabaja.

Posteriormente, en el capítulo 2, se explica el modelo físico-matemático con el que se obtiene la solución numérica. En el capítulo 3 se explica las mejoras que se añaden al código existente, en la forma de la condición de contorno que simula los efectos de la convección natural y la preiteración.

Finalmente, en el capítulo 4, se realizará un análisis de los resultados.

Adicionalmente se incluyen en un anexo el código elaborado de la condición de contorno y la preiteración.

LISTAS DE FIGURAS

Figura 1.1 Tipos de pilas de combustible (Cambridge, 2020)	8
Figura 1.2: Reacción electroquímica	8
Figura 1.3: Partes de una pila de combustible PEM (Barbir, 2012).....	9
Figura 1.4: Curva típica de pérdidas por activación (Barbir, 2012).....	11
Figura 1.5: Curva típica de pérdidas óhmicas (Barbir, 2012)	11
Figura 1.6: Curva típica de pérdidas por pérdida de concentración (Barbir, 2012).....	12
Figura 1.7: Curvas típicas de las pérdidas (Barbir, 2012)	13
Figura 1.8 Curva de Polarización (Bouaicha, Allagui, Aglzim, Rouane, & Mami, 2016)	13
Figura 1.9: Estructura general de OpenFOAM (OpenCFD Limited, 2018).....	14
Figura 1.10: Estructura de un caso en OpenFOAM (OpenCFD Limited, 2018)	15
Figura 3.1 Esquema de la convección (Velasco, 2017).....	21
Figura 3.2: Cara superior de la pila.....	23
Figura 3.3: Cara inferior de la pila	23
Figura 4.1: Distribución de temperatura en la superficie exterior de la celda	27
Figura 4.2: Distribución de temperatura en la sección central	27
Figura 4.3: Grafica de evolución de temperaturas.....	28
Figura 4.4: Distribución de temperatura en el canal de oxígeno	29
Figura 4.5: Distribución de temperatura en el canal de hidrógeno	29
Figura 4.6: Comparación de tiempos de simulación	31

LISTA DE TABLAS

Tabla 1.1 Energía libre de Gibbs (Barbir, 2012).....	10
Tabla 4.1: Comparación de resultados.....	30

INDICE

DECLARACION DE AUTORIA Y ORIGINALIDAD	2
RESUMEN	3
LISTAS DE FIGURAS.....	4
LISTA DE TABLAS.....	5
1 INTRODUCCIÓN Y OBJETIVOS	7
1.1 Pila de combustible	8
1.2 OpenFOAM.....	14
2 MODELO FÍSICO-MATEMÁTICO.	16
2.1 Modelo de simulación.....	16
2.2 Método iterativo del modelo.....	20
3 APORTACIONES AL MODELO.....	21
3.1 Condición de contorno.....	21
3.2 Preiteración.....	25
4 SIMULACIÓN.....	27
5 CONCLUSIONES.	32
6 BIBLIOGRAFÍA.....	33
7 Anexos.....	34
7.1 Código de la condición de contorno.	34
7.2 Código de la preiteración.	40

1 INTRODUCCIÓN Y OBJETIVOS

La Dinámica de Fluidos Computacional o CFD (*Computational Fluid Dynamics*), es una disciplina que se sirve de la capacidad de procesamiento de los ordenadores para la resolución de problemas de mecánica de fluidos mediante el cálculo numérico y permitiendo a su vez representar gráficamente en entornos 3D las soluciones obtenidas.

En este trabajo se va a realizar un estudio termodinámico de una Pila de combustible, empleando un software CFD de código abierto denominado OpenFOAM.

Se busca implementar mejoras en el código existente, desarrollado por el grupo de investigación de simulación computacional del LIFTEC, para ello se incorporará un módulo de cálculo previo o preiteración que permita obtener una temperatura inicial cercana a la de funcionamiento, acertando así el tiempo de simulación: cuanto más cerca de la temperatura final nos situemos al comienzo de la simulación más rápido convergerá el cálculo.

Posteriormente se implementará una condición de contorno que simule la disipación de calor por convección natural que se produce cuando la pila se encuentra sumergida en un aceite refrigerante y analizaremos como se distribuye el calor dentro de la pila, este modelo supondrá una mejora substancial respecto al modelo actual, en el que se supone una temperatura fija en el entorno, siendo esta nueva condición de convección natural una aproximación más cercana a la realidad.

En esta introducción se va a explicar:

- Qué es una Pila de combustible.
- Qué es OpenFOAM.

1.1 Pila de combustible

Una pila de combustible es un convertidor de energía electroquímica, que convierte la energía química del combustible en corriente eléctrica continua. Esta transformación de energía ocurre por la oxidación del combustible, generalmente hidrógeno, en el ánodo y reducción de oxígeno en el cátodo.

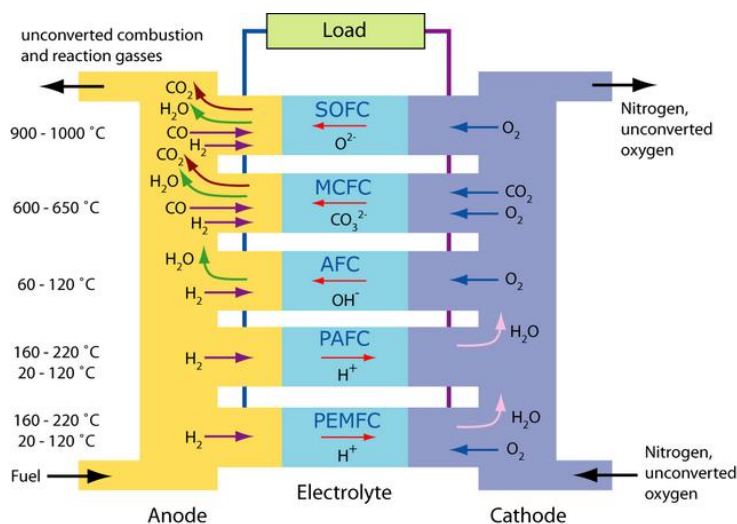


Figura 1.1 Tipos de pilas de combustible (Cambridge, 2020)

En este trabajo nos centraremos en las pilas de membrana de intercambio protónico PEM (*Proton Exchange Membrane*), en concreto en las pilas HTPEM (*High Temperature Proton Exchange Membrane*) cuya principal diferencia con las PEM es que su temperatura de trabajo es mayor, entre 100 y 200 °C.

En las pilas PEM la reacción electroquímica es la siguiente:

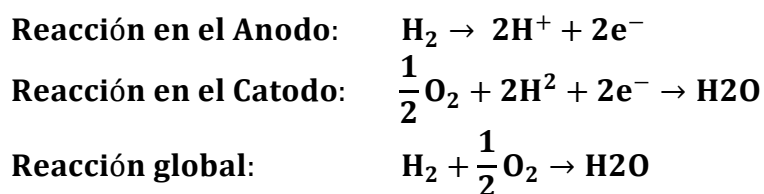


Figura 1.2: Reacción electroquímica

Como combustible se usa hidrógeno y como agente oxidante oxígeno, resultando en un flujo continuo de electrones entre el ánodo y el cátodo mientras se disponga de oxidante y combustible.

Las pilas PEM constan de varios componentes, en el centro se sitúa la membrana polimérica, a cada lado de esta hay un electrodo.

Estos electrodos tienen que ser porosos para permitir el paso de los gases hacia la capa catalítica que hay entre la membrana y los electrodos, que es donde se producen

las reacciones electroquímicas. Este conjunto se suele denominar MEA (*Membrane Electrode Assembly*).

El MEA se encuentra entre la capa de difusión de gases, GDL (*Gas Diffusion Layer*), estas capas proporcionan vías para la difusión de los gases reactivos hacia la capa catalítica y la evacuación de los productos, también conecta la capa catalítica con la placa bipolar.

Las placas bipolares unen eléctricamente las celdas con las adyacentes completando el circuito eléctrico y separan los flujos de gases que circulan por las celdas adyacentes.

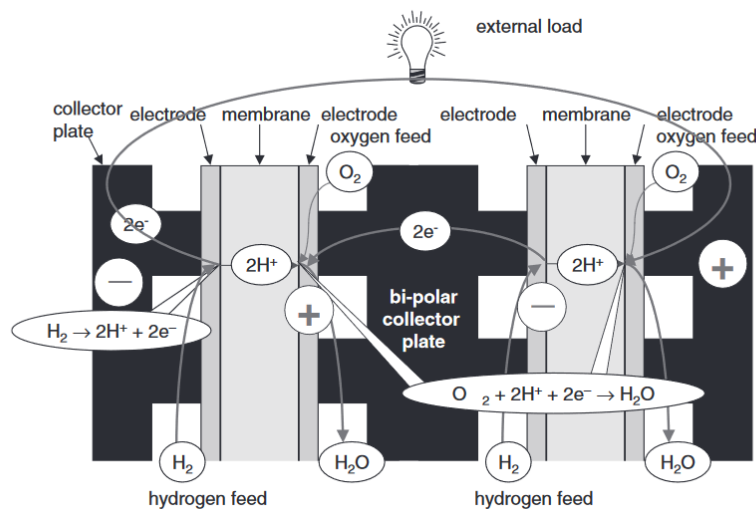


Figura 1.3: Partes de una pila de combustible PEM (*Barbir, 2012*)

En resumen, las partes principales de una pila de combustible son:

- Membrana de intercambio protónico
- Electrodo – Capa catalítica
- Capa de difusión de gases
- Placa bipolar

Curva de polarización.

Las reacciones electroquímicas que ocurren en la pila generan un flujo de electrones y por tanto corriente eléctrica el cerrarse el circuito eléctrico. A través de la curva de polarización se relaciona la densidad de corriente con la diferencia de potencial que se puede obtener entre los electrodos.

En esta curva se relacionan todas las pérdidas que se producen con el potencial máximo que podría obtenerse, las principales pérdidas de potencial que se producen son pérdidas por activación, pérdidas por pérdida de concentración y óhmicas.

El trabajo eléctrico se define como:

$$W_{ele} = nFE \quad (1.1)$$

Donde n es el número de electrones intercambiado por molécula de H₂, F es la constante de Faraday y E es el potencial de la pila.

	ΔG (kJ mol ⁻¹)
$H_2 + \frac{1}{2}O_2 \rightarrow H_2O(l)$	-237.34
$H_2 + \frac{1}{2}O_2 \rightarrow H_2O(g)$	-228.74

Tabla 1.1 Energía libre de Gibbs (Barbir, 2012)

La máxima energía teórica que se puede generar corresponde a la energía libre de Gibbs, ΔG :

$$W_{ele} = -\Delta G \quad (1.2)$$

El potencial teórico máximo de una pila de combustible o potencial de Nernst, E_0 es:

$$E_0 = \frac{-\Delta G}{nF} = \frac{228740 \frac{J}{mol}}{2 * 96485 \frac{C}{mol}} = 1.18 V \quad (1.3)$$

Cuando la pila trabaja con el circuito eléctrico abierto, sin conectar al resto del circuito eléctrico, el potencial que se obtiene es menor el potencial teórico.

- Perdidas por activación.

Estas pérdidas están asociadas a la lentitud de las reacciones, que requieren un cierto diferencial de potencial o sobrepotencial para iniciarse, siendo mucho mayores en la reducción del oxígeno. A partir de la ecuación de Butler–Volmer simplificada obtenemos una expresión del potencial de activación en función de la densidad de corriente:

$$\text{En el catodo: } \Delta V_{act,c} = \frac{RT}{\alpha_c F} * \ln \left(\frac{i}{i_{0,c}} \right) \quad (1.4)$$

$$\text{En el anodo; } \Delta V_{act,a} = \frac{RT}{\alpha_a F} * \ln \left(\frac{i}{i_{0,a}} \right) \quad (1.5)$$

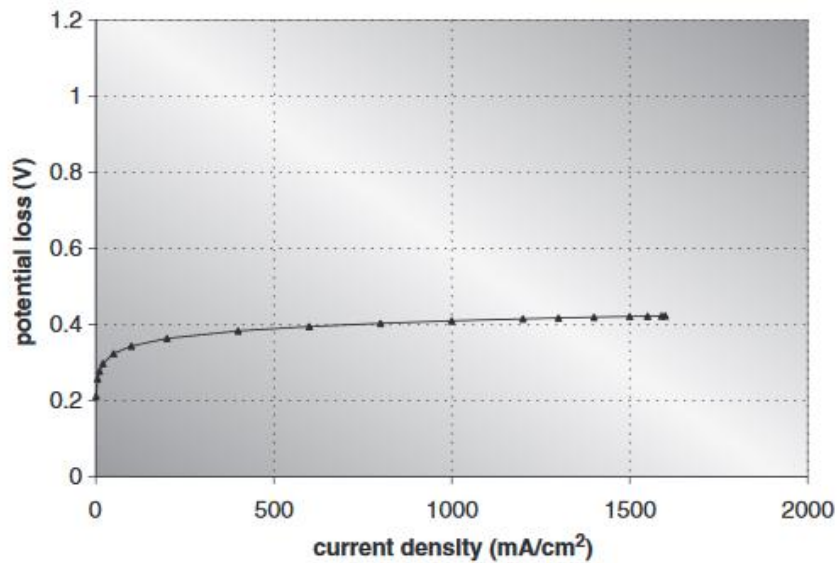


Figura 1.4: Curva típica de pérdidas por activación (Barbir, 2012)

Siendo, α_c y α_a coeficientes de transferencia, $i_{0,c}$ e $i_{0,a}$ la densidad de corriente de referencia en el cátodo y ánodo respectivamente, R la constante de los gases ideales.

- Perdidas óhmicas

Las pérdidas óhmicas ocurren por la resistencia al flujo de iones en el electrolito y al flujo de electrones por los componentes conductores eléctricos. Esta pérdida de potencial se puede calcular a partir de la ley de Ohm.

$$\Delta V_{ohm} = i * R_i \tag{1.6}$$

Siendo R_i la suma de las diferentes resistencias internas, iónica, electrónica y constante.

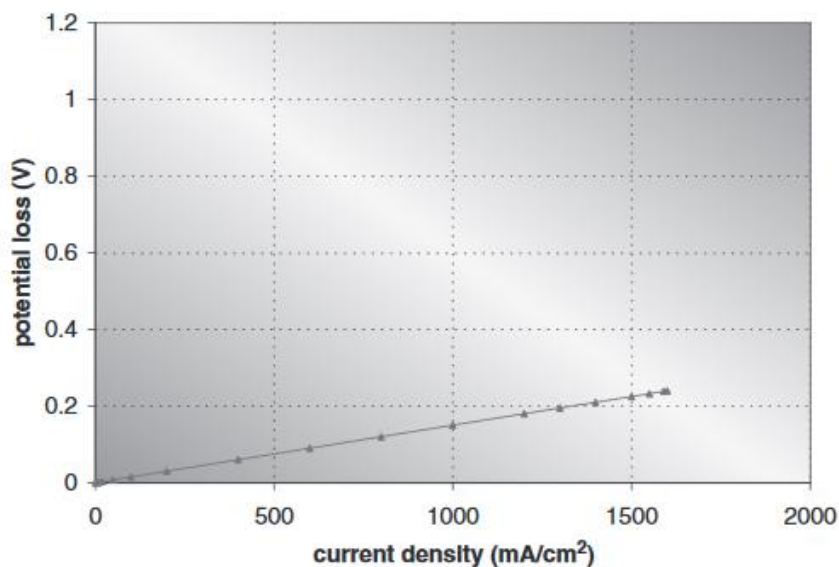


Figura 1.5: Curva típica de pérdidas óhmicas (Barbir, 2012)

- Perdidas por pérdida de concentración

Las pérdidas por pérdida de concentración suceden cuando se consume rápidamente los reactivos por las reacciones electroquímicas, esto supone un mayor problema cuando se opera con altas intensidades, donde se requiere una mayor concentración de los reactivos para mantener las reacciones.

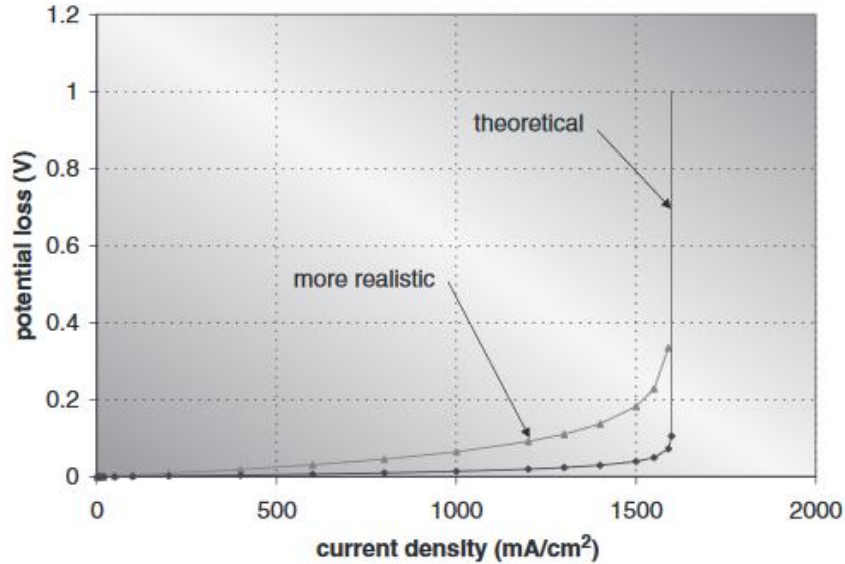


Figura 1.6: Curva típica de pérdidas por pérdida de concentración (Barbir, 2012)

Las pérdidas por pérdida de concentración pueden expresarse con la siguiente ecuación:

$$\Delta V_{conc} = \frac{RT}{nF} * \ln\left(\frac{i_L}{i_L - i}\right) \quad (1.7)$$

Siendo i_L la densidad de corriente limitante, que es la intensidad límite que podría producir la pila ya que no habría una concentración suficiente de los reactivos en la capa catalítica. Esta intensidad no se suele alcanzar en pilas reales por lo que una expresión empírica describe mejor las pérdidas por pérdida de concentración.

$$\Delta V_{conc} = c * \exp\left(\frac{i}{d}\right) \quad (1.8)$$

Donde c y d son parámetros que se obtienen experimentalmente.

La expresión que describe la curva de polarización sería la siguiente:

$$V_{cell} = E_r - (\Delta V_{act} + \Delta V_{conc})_a - (\Delta V_{act} + \Delta V_{conv})_c - \Delta V_{ohm} \quad (1.9)$$

E_r es el potencial de equilibrio. Introduciendo las ecuaciones de cada tipo de pérdidas quedaría de la siguiente forma:

$$E_{cell} = E_r - \frac{RT}{\alpha_c F} \ln\left(\frac{i}{i_{o,c}}\right) - \frac{RT}{\alpha_a F} \ln\left(\frac{i}{i_{o,a}}\right) - \frac{RT}{nF} \ln\left(\frac{i_{L,c}}{i_{L,c} - i}\right) - \frac{RT}{nF} \ln\left(\frac{i_{L,a}}{i_{L,a} - i}\right) - iR_i \quad (1.10)$$

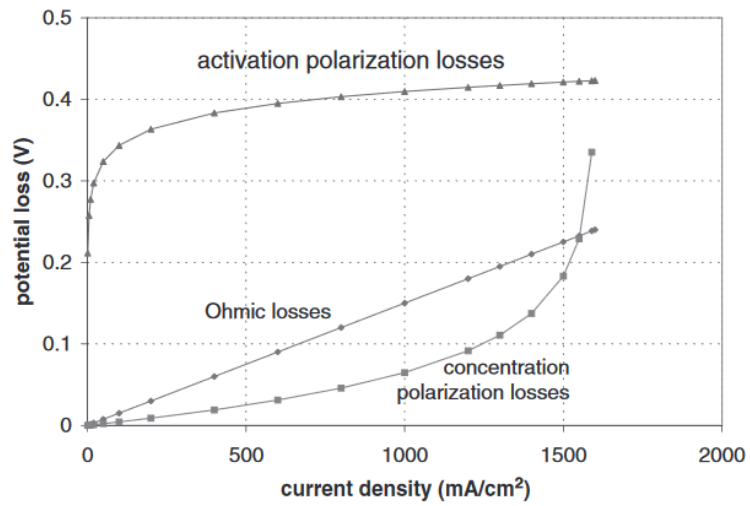


Figura 1.7: Curvas típicas de las pérdidas (Barbir, 2012)

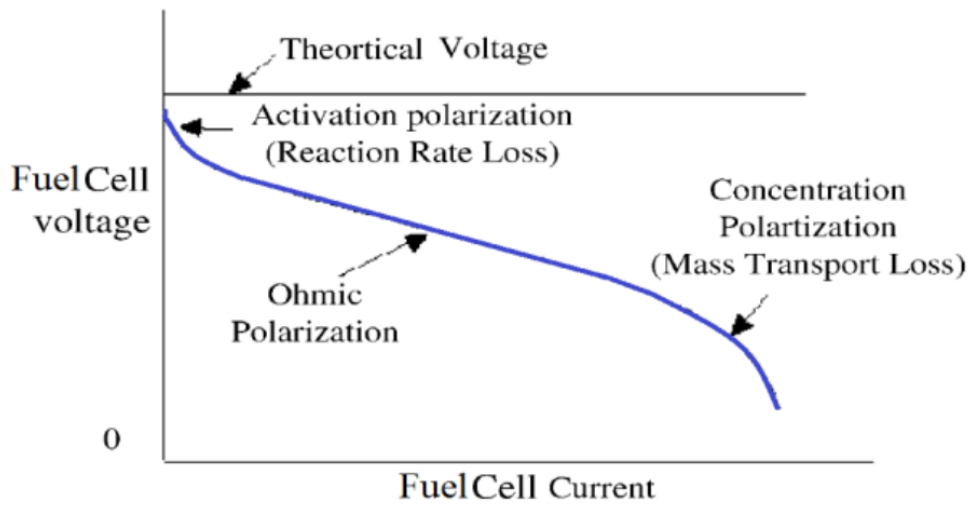


Figura 1.8 Curva de Polarización (Bouaicha, Allagui, Agzim, Rouane, & Mami, 2016)

1.2 OpenFOAM

OpenFOAM (*Open Source Field Operation and Manipulation*) es un software CFD de código abierto basado en el lenguaje de programación C++. En la actualidad es un software muy empleado en muchas áreas de la ciencia e ingeniería. Su aplicación principal es en el campo de la mecánica de fluidos, aunque también puede abarcar otros campos como por ejemplo el electromagnetismo o la mecánica de sólidos.

OpenFOAM es una biblioteca C++, en la que se crean ficheros ejecutables conocidos como aplicaciones. Hay dos tipos de aplicaciones: *solvers* que se encargan de resolver las ecuaciones que gobiernan el problema y *utilities*, que se encargan de tareas relacionadas con la manipulación de los datos.

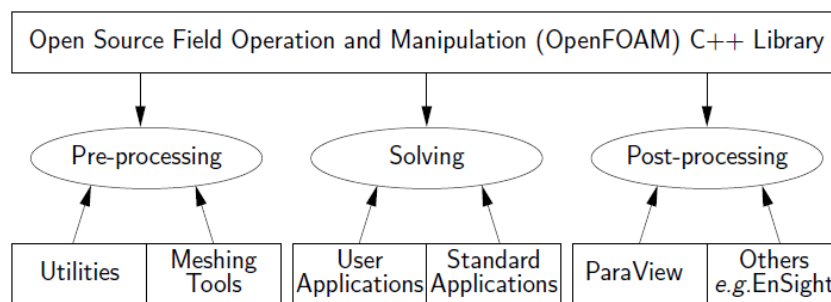


Figura 1.9: Estructura general de OpenFOAM (*OpenCFD Limited, 2018*)

Como se ve en la Figura 1.9 la estructura de trabajo con OpenFOAM se divide en tres partes, preprocesamiento, resolución y postprocesamiento.

- **Preprocesamiento**

El preprocesamiento consiste en crear el modelo que se desea utilizar. Se suele partir de modelos y geometrías previamente existentes y modificándolo a nuestra conveniencia.

Toda la información y archivos que describen un problema específico y la forma en la que se va a resolver se agrupa en un conjunto de directorios llamado caso. Este caso se estructura en tres carpetas: *system*, *constant* y una carpeta temporal inicial.

En la carpeta *constant* se encuentran los archivos que contienen toda la información del mallado de la geometría, que se suele generar con el *utility blockMesh* y las propiedades físicas necesaria para ejecutar el *solver*.

En la carpeta *system* se encuentra los archivos que contienen la información sobre la simulación y el proceso de resolución, como los tiempos de simulación, el incremento de tiempo de cada simulación, el intervalo de tiempo prescrito para guardar datos, los esquemas de discretización utilizadas en la resolución, las tolerancias, los solucionadores de ecuaciones y otros algoritmos de control.

En la carpeta temporal inicial se establecen las condiciones iniciales de las variables de interés y sus correspondientes condiciones de contorno.

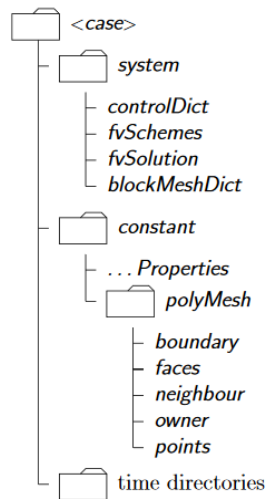


Figura 1.10: Estructura de un caso en OpenFOAM (OpenCFD Limited, 2018)

- **Ejecución**

Para resolver un problema concreto se debe ejecutar un *solver*, el cual contiene las ecuaciones que describen el problema que se quiere simular y los algoritmos que las resuelven numéricamente.

- **Postprocesamiento**

Una vez concluida la simulación dentro de la carpeta del caso se crean varias carpetas temporales que contienen los valores calculados de las variables, para visualizar estos resultados se empleará un *software* de visualización, siendo el más usado *Paraview*.

Paraview dispone de una interfaz gráfica que nos permite el visualizado de los resultados con un código de colores y diversas funciones para su análisis.

2 MODELO FÍSICO-MATEMÁTICO.

En este apartado se explicarán el modelo empleado para la simulación, la condición de contorno que usara para modelar el efecto de la convección natural en las paredes de la pila y la preiteración que calculara las condiciones de temperatura aproximadas al inicio de la simulación.

2.1 Modelo de simulación

El modelo que se va a usar para realizar las simulaciones está basado en el modelo desarrollado por el grupo de investigación de simulación computacional del LIFTEC.

La pila de combustible estará formada por las siguientes zonas volumétricas: interconectores, cátodo, ánodo, capa funcional del cátodo, capa funcional del ánodo, electrolito, entrada y salida de aire y entrada y salida de fuel.

Existen en la literatura científica diversos niveles de aproximación de la descripción de los fenómenos de transporte en una pila de combustible polimérica. Es fundamental ser equilibrado y consistente con el nivel de detalle escogido. Por ello se realizan ciertas simplificaciones respecto al modelo real, debido a la complejidad de este, estas hipótesis serán:

- Al trabajar con una pila de alta temperatura HTPEM, no habrá agua líquida en el dominio.
- Las reacciones electroquímicas ocurren en la interfaz entre los electrodos y el electrolito. Para ello se utiliza un *modelo de capa catalítica infinitamente delgada*, en el que se añade como condición de contorno que el consumo y producción de las especies se modela como la “desaparición” y “aparición” de estas en un lado u otro.
- En el lado del aire se supone una mezcla de oxígeno, nitrógeno y agua, en el lado del fuel se supone una mezcla de hidrogeno y agua.
- Se considera mezcla de gases perfectos.
- Se considera que la pila está en estado estacionario.

Las ecuaciones que se usan para modelar la pila son la siguientes:

- Ecuaciones de transporte
 - Ecuación de continuidad

$$\nabla * (\rho u) = 0 \tag{2.1}$$

- Ecuación de conservación del momento

$$\frac{1}{\varepsilon^2} \nabla * (\rho u u) = -\nabla p + \frac{1}{\varepsilon} \nabla * (\mu * \nabla u) + S_p \quad (2.2)$$

- Ecuación de conservación de las especies.

$$\nabla * (\rho u y_i) = \nabla * (\Gamma_{eff} * \nabla y_i) \quad (2.3)$$

- Ecuación de conservación de energía

$$\nabla * (\rho u c_p T) = \nabla * (k_{eff} \nabla (c_p T)) + S \quad (2.4)$$

Donde ρ es la densidad de los gases, u la velocidad de los gases, p es la presión de los gases, μ es la viscosidad dinámica de los gases, S_p es el termino fuente en el medio poroso, y_i es la fracción molar de cada especie, Γ_{eff} coeficiente de difusividad, c_p es el calor especifico de los materiales la pila, T es la temperatura a la que se encuentran los materiales la pila, k_{eff} es la conductividad térmica de los materiales la pila, S es el termino fuente de calor.

- Termino fuente en el medio poroso, que se aplica en el cátodo, ánodo y sus capas funcionales.

$$S_p = -\frac{\mu u}{k_D} \quad (2.5)$$

Donde k_D es la permeabilidad del medio poroso.

- Condición de contorno en el ánodo:
 - Condición de contorno de desaparición de la especie de O2:

$$\dot{S} = \dot{m}_{O_2} * (1 - y_{O_2}) \quad (2.6)$$

- Donde \dot{m}_{O_2} es la condición de contorno de desaparición de masa de O2:

$$\dot{m}_{O_2} = -\frac{M_{O_2} i}{nF} \quad (2.7)$$

Aquí n es el numero de electrones intercambiado por molécula de O2

- Condición de contorno de la especie N2:

$$\dot{S} = -\dot{m}_{O_2} * y_{N_2} \quad (2.8)$$

- Condición de contorno en el cátodo:
 - Condición de contorno desaparición de la especie H2 debido al consumo de H2

$$\dot{S} = \dot{m}_{H_2} (1 - y_{H_2}) \quad (2.9)$$

- Condición de contorno de aparición de la especie H₂O debido al consumo de H₂

$$\dot{S} = -\dot{m}_{H_2} y_{H_2O} \quad (2.10)$$

- Condición de contorno de aparición de la especie H₂O debido a la producción de H₂O

$$\dot{S} = \dot{m}_{H_2O} (1 - y_{H_2O}) \quad (2.11)$$

- Condición de contorno de desaparición de la especie H₂ debido a la producción de H₂O

$$\dot{S} = -\dot{m}_{H_2O} y_{H_2} \quad (2.12)$$

- Donde \dot{m}_{H_2} es la condición de contorno de desaparición de masa de H₂:

$$\dot{m}_{H_2} = -\frac{M_{H_2} i}{nF} \quad (2.13)$$

- Donde \dot{m}_{O_2} es condición de contorno de aparición de masa de H₂O:

$$\dot{m}_{H_2O} = \frac{M_{H_2O} i}{nF} \quad (2.14)$$

Siendo i la densidad de corriente, M_{O_2} , M_{H_2} y M_{H_2O} las masas molares de O₂, H₂ y H₂O.

- Electroquímica:

- Voltaje de la pila

$$V = E - iR_{membrana} - \eta_a - \eta_c \quad (2.15)$$

Donde η_a y η_c son los sobrepotenciales anódico y catódico, $R_{membrana}$ es la resistencia de la membrana al paso de los protones.

- Potencial de Nernst

$$E = E_0 + \frac{RT}{2F} \ln \left(\frac{x_{H_2} x_{O_2}^{0.5}}{X_{H_2O}} \right) + \frac{RT}{4F} \ln \left(\frac{p}{p_0} \right)_a \quad (2.16)$$

$$E_0 = -\frac{\Delta G_0}{nF} \quad (2.17)$$

Donde E es el potencial de Nernst, p/p_0 es cociente entre la presión en el lado del aire con la presión del aire al que se evalúa E_0 , ΔG_0 es la energía libre de Gibbs.

- Sobrepotencial de activación

$$i_i^{ref} = i_{Tref}^{ref} \left(-\frac{E_i}{R} \left(\frac{1}{T} - \frac{1}{T_{ref}} \right) \right) \quad (2.18)$$

Donde i_i^{ref} es la densidad de corriente de referencia del ánodo (i_a^{ref}) o del cátodo (i_c^{ref}), i_{Tref}^{ref} es la densidad de corriente de referencia a la temperatura de referencia, E_i es la energía de activación en el ánodo (E_a) o en el cátodo (E_c), T_{ref} es la temperatura de referencia (353 K), T es la temperatura a la que se encuentra.

Para el cálculo de los sobrepotenciales de activación usaremos modelos diferentes para el ánodo y el cátodo. En el ánodo usaremos una ecuación modificada a partir de la ecuación de Butler-Volmer:

$$i_a = i_a^{ref} \left(\frac{P_{fuel} X_{H_2}}{P_{atm}} \right)^a \left(-\exp\left(\frac{-\alpha_a F \eta_a}{RT}\right) + \exp\left(\frac{(2 - \alpha_a) F \eta_a}{RT}\right) \right) \quad (2.19)$$

Donde α_a es el coeficiente de transferencia del ánodo, P_{fuel} es la presión a la que se encuentra el fuel, P_{atm} es la presión atmosférica, X_{H_2} es la fracción molar del hidrogeno, a es un coeficiente de dependencia de la presión y η_a es el sobrepotencial de activación en el ánodo.

En el cátodo usaremos un modelo aglomerado que incorpora la influencia de las perdidas por concentración (Li, Yuan, Xie, & Sundén, 2018):

$$i_c = CL_{thiknes} n F P_{O_2} H_{O_2} \left(\frac{1}{E_r k_c (1 - \epsilon_{cl})} + \frac{(r_{agg} + \delta_i) \delta_i}{a_{agg} D_{O_2}} \right)^{-1} \quad (2.20)$$

$$k_c = \frac{i_c^{ref}}{n F (1 - \epsilon_{cl}) C_{O_2}^{ref}} \left(\exp\left(\frac{\alpha_c F \eta_c}{RT}\right) - \exp\left(\frac{-(1 - \alpha_c) F \eta_c}{RT}\right) \right) \quad (2.21)$$

Donde $CL_{thiknes}$ es el espesor de la capa catalítica, n es el número de electrones que se transfieren, P_{O_2} es la presión parcial del oxígeno, H_{O_2} es la constante de hidrogeno de Henry, k_c es la constante de velocidad de reacción, E_r es el factor de eficiencia del aglomerado, ϵ_{cl} es la fracción volumétrica del espacio vacío, r_{agg} radio del aglomerado, δ_i espesor de la capa ionomerica sobre el aglomerado, a_{agg} es el área efectiva del aglomerado, D_{O_2} es la difusividad del oxígeno en el ionómero, $C_{O_2}^{ref}$ es la concentración de referencia, α_c es el coeficiente de transferencia del cátodo y η_c es el sobrepotencial de activación en el cátodo.

- Termina Fuente de calor en el electrolito

El calor generado se calcula como la suma de la diferencia de entropía en las reacciones más el calor generado por el paso de los protones por la membrana

$$\dot{S} = \frac{i^2}{K_p} + i * \left(\eta_a + \eta_c + \frac{T * \Delta S}{n * F} \right) \quad (2.22)$$

Donde \dot{S} es el termino fuente del calor, K_p es la conductividad protónica de la membrana, ΔS es la variación de entropía.

$$\Delta S = S_{O_2}^0 + S_{H_2}^0 - S_{H_2O}^0 + \int_T^{T_{out}} \frac{C p_{O_2}}{T} dT + \int_T^{T_{out}} \frac{C p_{H_2}}{T} dT - \int_T^{T_{out}} \frac{C p_{H_2O}}{T} dT + R * \ln \left(\frac{X_{O_2}^{0.5} * X_{H_2}}{X_{H_2O}} \right) \quad (2.23)$$

2.2 Método iterativo del modelo.

El objetivo de este proceso iterativo es acoplar las ecuaciones físicas y electroquímicas y para ello se utiliza el valor de la densidad de corriente. Para ello es necesario que el valor de intensidad con el que se fijan las condiciones de contorno en la resolución de las ecuaciones físicas converja con el valor de intensidad calculado a partir de las ecuaciones electroquímicas.

Primero es necesario fijar el valor de tensión al que se realiza la simulación. Además, es necesario inicializar el proceso fijando un valor de intensidad i que sirva de punto de partida.

1. A partir del valor de i se calculan las condiciones de contorno en las capas catalíticas para la velocidad y las concentraciones, ecuaciones (2.6) a (2.14).
2. Con el valor actualizado de las condiciones de contorno se resuelven las ecuaciones de continuidad, conservación del momento, conservación de las especies y conservación de energía, ecuaciones (2.1), (2.2), (2.3) y (2.4). Es decir, se obtienen las distribuciones de presión, velocidad, temperatura y concentraciones de especies en los dominios fluidos.
3. Utilizando el valor de i y las propiedades de las especies en las capas catalíticas se obtienen los sobrepotenciales según las ecuaciones electroquímicas, ecuaciones (2.19) y (2.20). También se calcula el potencial reversible de Nernst ecuación (2.16).
4. Se obtiene el nuevo valor de intensidad de acuerdo a los valores de potenciales calculados:

$$i_{calculada} = \frac{(E - \eta_a - \eta_c - V)}{R_{membrana}} \quad (2.24)$$

5. Se compara el valor de $i_{calculada}$ con el valor de la usada para fijar las condiciones de contorno.
- Si la diferencia es superior al valor fijado para la convergencia se actualiza el valor de la intensidad aplicando un valor de relajación y se vuelve al punto 1.

$$i_{new} = i + (i_{calculada} - i) * relajacion \quad (2.25)$$

- Si se cumple el criterio de convergencia la simulación concluye.

$$\frac{i_{new} - i}{i} < 0.001 \quad (2.26)$$

3 APORTACIONES AL MODELO

3.1 Condición de contorno.

Dado la imposibilidad de simular el fluido en el que se encuentra sumergida la pila, ya que esto añadiría una complejidad y coste computacional excesivos, para el análisis termodinámico que se quiere realizar se opta por implementar en el modelo una condición de contorno en las superficies exteriores de la pila que simule la disipación de calor por convección natural.

Para esto nos serviremos de la biblioteca groovyBC de OpenFOAM que nos permite incorporar directamente las ecuaciones que describen la condición de contorno. GroovyBC implementa en OpenFOAM la condición de contorno tipo Robin.

La disipación de calor por convección se produce de forma similar a la representada en la Figura 3.1

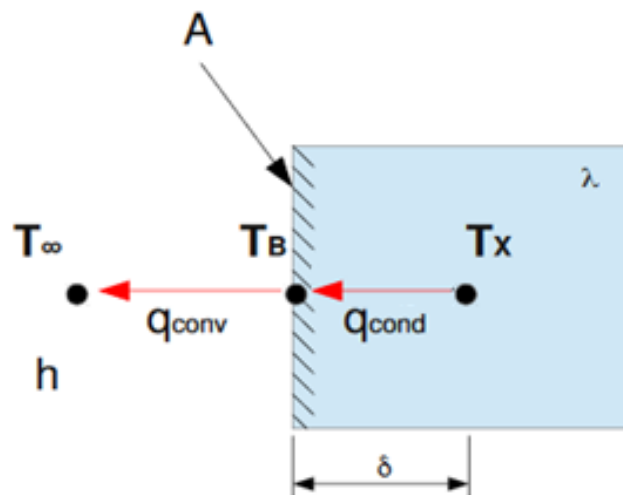


Figura 3.1 Esquema de la convección (Velasco, 2017)

Este proceso físico se describe con las siguientes ecuaciones:

$$q_{conv} = h * A * (T_\infty - T_B) \quad (3.1)$$

$$q_{cond} = \frac{k}{\delta} * A * (T_B - T_x) \quad (3.2)$$

$$q_{conv} = q_{cond} \quad (3.3)$$

$$\frac{q}{A} = h * (T_\infty - T_B) = \frac{k}{\delta} * (T_B - T_x) \quad (3.4)$$

Donde q_{conv} y q_{cond} son los calores disipados por convección y conducción, respectivamente, A es el área de la superficie por la que se produce el intercambio de calor, k es el coeficiente de conducción de la superficie, h el coeficiente de convección del fluido, δ es la distancia del centro de la celda al centro de la superficie y T_x , T_B y T_∞ las temperaturas en el centro, borde y en el “infinito”.

En OpenFOAM los valores en los bordes se calculan con la siguiente expresión:

$$X_B = f * X_{ref} + (1 - f) * \left(X_x + \frac{Grag_{ref}(X)}{deltaCoeffs} \right) \quad (3.5)$$

Siendo X_{ref} un valor de referencia para la variable en la frontera, X_x el valor en el centro de la celda, $Grag_{ref}(X)$ el gradiente de referencia de la variable, $deltaCoeffs$ el inverso de la distancia entre el centro de la celda y el centro de la cara exterior y f es un factor que defina el tipo de condición de contorno, siendo tipo Dirichlet si $f = 1$, tipo Neumann si $f = 0$ o tipo Robin si $1 > f > 0$.

Para nuestro caso queremos que la ecuación (3.4) sea similar a (3.5), para ello despejaremos T_B .

$$T_B = T_\infty * \frac{h}{h + \frac{k}{\delta}} + T_x * \frac{\frac{k}{\delta}}{h + \frac{k}{\delta}} \quad (3.6)$$

En esta ecuación obtenemos el factor f y $(1 - f)$:

$$f = \frac{h}{h + \frac{k}{\delta}}; \quad (1 - f) = \frac{\frac{k}{\delta}}{h + \frac{k}{\delta}} \quad (3.7)$$

que reescribiéndolo quedan de la forma:

$$f = \frac{1}{1 + \frac{k}{\delta * h}} \quad (3.8)$$

Y la ecuación de la temperatura quedara:

$$T_B = T_\infty * \frac{1}{1 + \frac{k}{\delta * h}} + T_x * \left(1 - \frac{1}{1 + \frac{k}{\delta * h}} \right) \quad (3.9)$$

El siguiente paso es definir como se calcula el coeficiente de convección, en los diferentes casos que se nos presentan, que son varias superficies verticales, una superficie horizontal hacia arriba y otra hacia abajo, la principal diferencia entre estos será la formula usada para el cálculo del número de Nusselt. En cada caso habrá también ecuaciones comunes, como la del coeficiente de convección y las de los números adimensionales Rayleigh y Prandtl.

Coeficiente de convección:

$$h = Nu * \frac{k_f}{L} \quad (3.10)$$

Numero de Rayleigh:

$$Ra = \frac{g * \beta * (T_B - T_\infty) * L^3 * \rho * Cp}{\nu * k_f} \quad (3.11)$$

Numero de Prandtl:

$$Pr = \frac{\nu * \rho * Cp}{k_f} \quad (3.12)$$

Siendo L la longitud característica de la superficie de convección de cada caso, g es la aceleración de la gravedad, k_f el coeficiente de conducción del fluido, β es el coeficiente de expansión térmica del fluido, ρ es la densidad del fluido, ν es la viscosidad cinemática del fluido, Cp es el calor específico del fluido.

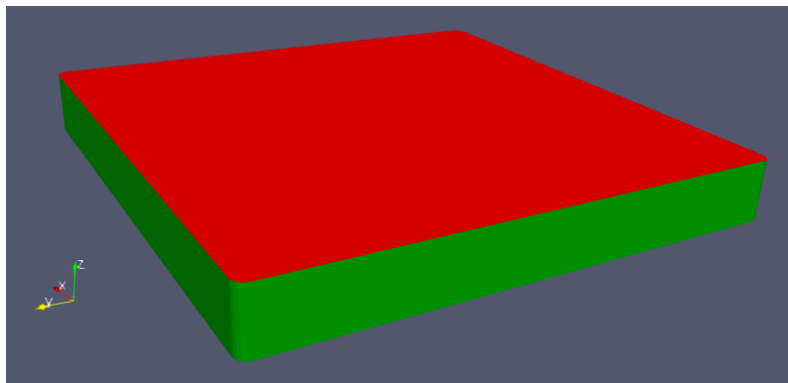


Figura 3.2: Cara superior de la pila

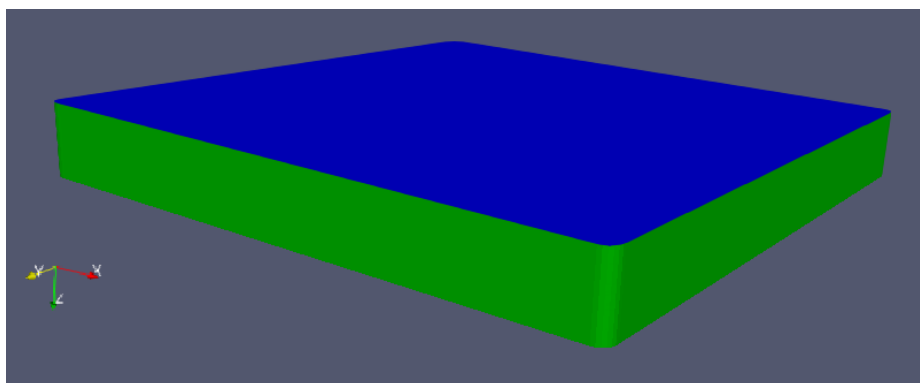


Figura 3.3: Cara inferior de la pila

Para una superficie vertical, zona de color verde en la Figura 3.2 , el número de Nusselt lo calculamos como:

$$Nu = \left(0.825 + \frac{0.387 * Ra^{\frac{1}{6}}}{\left(1 + \left(\frac{0.5}{Pr} \right)^{\frac{9}{16}} \right)^{\frac{8}{27}}} \right)^2 \quad (3.13)$$

Y la longitud característica en una superficie vertical será su altura.

Para la superficie horizontal hacia arriba, zona de color rojo en la Figura 3.2 , calculamos Nusselt como:

$$Nu = 0.54 * Ra^{1/4} \quad (3.14)$$

Para la superficie horizontal hacia abajo, zona de color azul en la Figura 3.3, calculamos Nusselt como:

$$Nu = 0.27 * Ra^{1/4} \quad (3.15)$$

En ambas superficies horizontales la longitud característica se define como el cociente entre al área y el perímetro de la superficie.

$$L = \frac{Area}{Perimetro} \quad (3.16)$$

El fluido que se ha elegido es un aceite para transferencia de calor CHEVRON HEAT TRANSFER OIL grado 46 (Chevron Lubricants, 2015), a partir de las tablas de valores de propiedades del fabricante se hace una regresión cuadrática y obtenemos las siguientes ecuaciones que relacionan cada propiedad con la temperatura.

$$\beta = 4 * 10^{-9} * T_f^2 - 1,448 * 10^{-6} * T_f + 8,31505 * 10^{-4} \quad (3.17)$$

$$\rho = -3,5 * 10^{-4} * T_f^2 - 0.400994 * T_f + 1011.57359 \quad (3.18)$$

$$v = 5 * 10^{-10} * T_f^2 - 4,642 * 10^{-7} * T_f + 1,110221 * 10^{-4} \quad (3.19)$$

$$k_f = -0,00014 * T_f + 0,16984 \quad (3.20)$$

$$Cp = -0,00320 * T_f^2 + 6,85252 * T_f + 209,00043 \quad (3.21)$$

Estas propiedades se evalúan a la llamada temperatura de película:

$$T_f = \frac{T_B + T_\infty}{2} \quad (3.22)$$

3.2 Preiteración

Puesto que el proceso iterativo tiene altas inercias, la diferencia entre las condiciones iniciales y el resultado final tiene mucha influencia en el tiempo de simulación, por tanto, con este cálculo previo se va a obtener una temperatura aproximada a la de operación.

Se plantea un balance de energía entre los calores que entran y los que salen. Los términos de este balance serán: el calor generado por las reacciones electroquímicas, el calor que aportan las especies gaseosas al entrar, el calor que extraen los productos a la salida y el calor que se disipara el fluido refrigerador por convección.

$$\sum Q = Q_{elec} + Q_{in} - Q_{out} - Q_{conv} \quad (3.23)$$

El calor que aportan los gases se calcula a partir de esta expresión:

$$Q_{in} = \sum \dot{m}_{in} * Cp_{in} * T_{in} \quad (3.24)$$

Donde \dot{m}_{in} es el flujo masico de entrada de cada especie, Cp_{in} es el calor especifico de cada especie y T_{in} la temperatura de entrada de los gases.

El calor que extraen los gases se calcula de forma similar:

$$Q_{out} = \sum \dot{m}_{out} * Cp_{out} * T_{out} \quad (3.25)$$

Donde T_{out} será la temperatura de los gases a la salida, \dot{m}_{out} el flujo masico de gases a la salida y Cp_{out} su calor especifico.

El calor generado por las reacciones electroquímicas se calcula con la siguiente ecuación:

$$Q_{elec} = \frac{i^2 * V_{electrolito}}{K_p} + i * \left(\eta_{anodo} + \eta_{catodo} + \frac{T_{out} * \Delta S}{2 * F} \right) * A_{electrolito} \quad (3.26)$$

Donde i es la densidad de corriente a la que se opera, $V_{electrolito}$ es el volumen del electrolito, K_p es la conductividad protónica del electrolito, η_{anodo} y η_{catodo} los sobrepotenciales en el ánodo y cátodo, T_{out} es la temperatura en el interior de la pila, ΔS es la variación de entropía y $A_{electrolito}$ el área del electrolito. ΔS se calcula con la ecuación (2.23). Los sobrepotenciales los calcula el programa mediante el método numérico iterativo de Ridder, que es un algoritmo de búsqueda de raíces.

Se toma T_{out} como la temperatura dentro de la pila como una aproximación cercana a la realidad, ya que la temperatura media en la pila es cercana a la temperatura a la que salen las especies.

El calor disipado por convección se calcula como:

$$Q_{conv} = \sum (h_i * A_{conv,i}) * (T_{out} - T_{\infty}) \quad (3.27)$$

Donde $A_{conv,i}$ es la superficie de cada cara que interviene en la convección, T_{∞} es la temperatura del fluido en el infinito, T_{out} es la temperatura a la que se encuentra la superficie de convección, h es el coeficiente de convección de cada superficie de convección, vertical, horizontal hacia arriba y hacia abajo.

Para obtener los coeficiente de convección usaremos las mismas ecuaciones que en la condición de contorno, ecuaciones (3.10) a (3.21).

A partir de estas ecuaciones planteamos un proceso iterativo cuyo objetivo es alcanzar una temperatura T_{out} que haga $\sum Q$ próximo a cero.

Primero fijaremos un valor de intensidad i al que va a realizar el cálculo y la temperatura T_{in} a la que entran las especies y calculamos el flujo masico de entrada \dot{m}_{in} de las especies, el consumo que se produce de estas a la intensidad i y el flujo masico de salida \dot{m}_{out} como la diferencia de estos.

Además, fijamos un valor de temperatura de salida T_{out} que sirva de punto de partida para la iteración.

Una vez dentro del bucle iterativo seguimos los siguientes pasos:

1. A partir de T_{out} calculamos el calor Q_{out} que evacuan las especies.
2. Calculamos los sobrepotenciales de activación, η_{anodo} y η_{catodo} , la variación de entropía ΔS y a partir de estos el calor generado por las reacciones electroquímicas.
3. Calculamos las propiedades del fluido a la temperatura de película, los parámetros de la convección y calor Q_{conv} que se disipa por la convección.
4. Calculamos el nuevo valor de la temperatura de salida $T_{calculada}$ que hará el sumatorio $\sum Q = 0$ con los valores que hemos calculado:

$$T_{calculada} = \frac{Q_{in} + Q_{elect} + T_{in}(h_{vertical} * A_{conv,V} + (h_{sup} + h_{inf}) * A_{conv,H})}{\sum \dot{m}_{out} * Cp_{out} + (h_{vertical} * A_{conv,V} + (h_{sup} + h_{inf}) * A_{conv,H})} \quad (3.28)$$

5. Comparamos el valor de $T_{calculada}$ con el valor usado en el bucle y se actualiza el valor de la temperatura aplicando un valor de relajación:

$$T_{new} = T_{out} + (T_{calculada} - T_{out}) * relajacion \quad (3.29)$$

6. Con los valores de calor obtenidos calculamos $\sum Q$.
- Si se cumple el criterio de convergencia el bucle concluye.

$$\sum Q < 0.001 \quad (3.30)$$

4 SIMULACIÓN

En este capítulo vamos a analizar los resultados obtenidos. Por un lado, la distribución de temperatura en la pila, y por otro, la influencia de la preiteración sobre el tiempo de computación.

Distribución de temperatura.

El software de visualización Paraview nos permite ver la distribución de temperatura en la geometría. Tomaremos un caso de ejemplo que trabaja a 2000 A/m^2 para ilustrar los resultados.

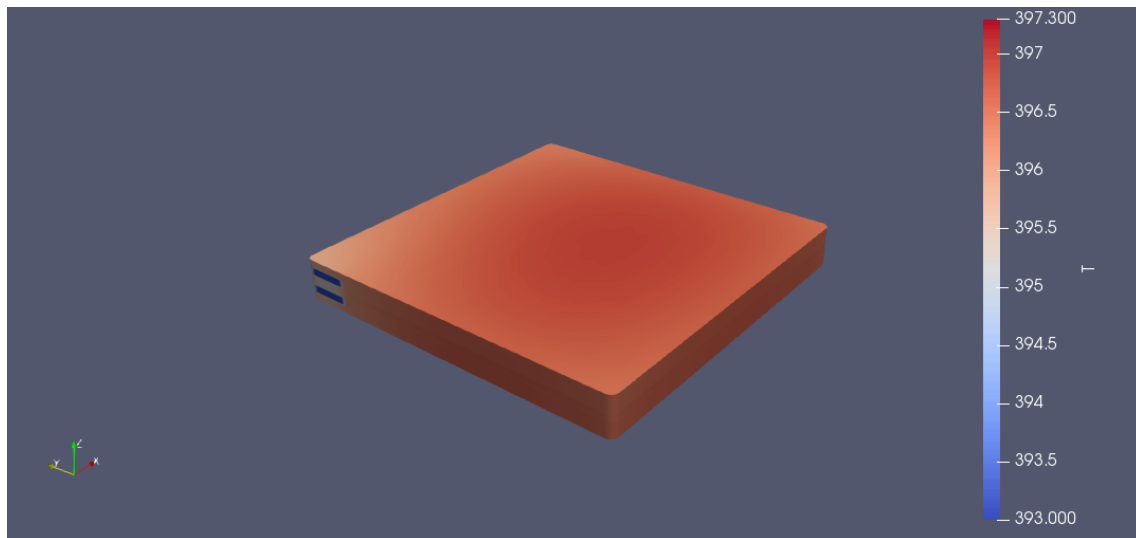


Figura 4.1: Distribución de temperatura en la superficie exterior de la celda



Figura 4.2: Distribución de temperatura en la sección central

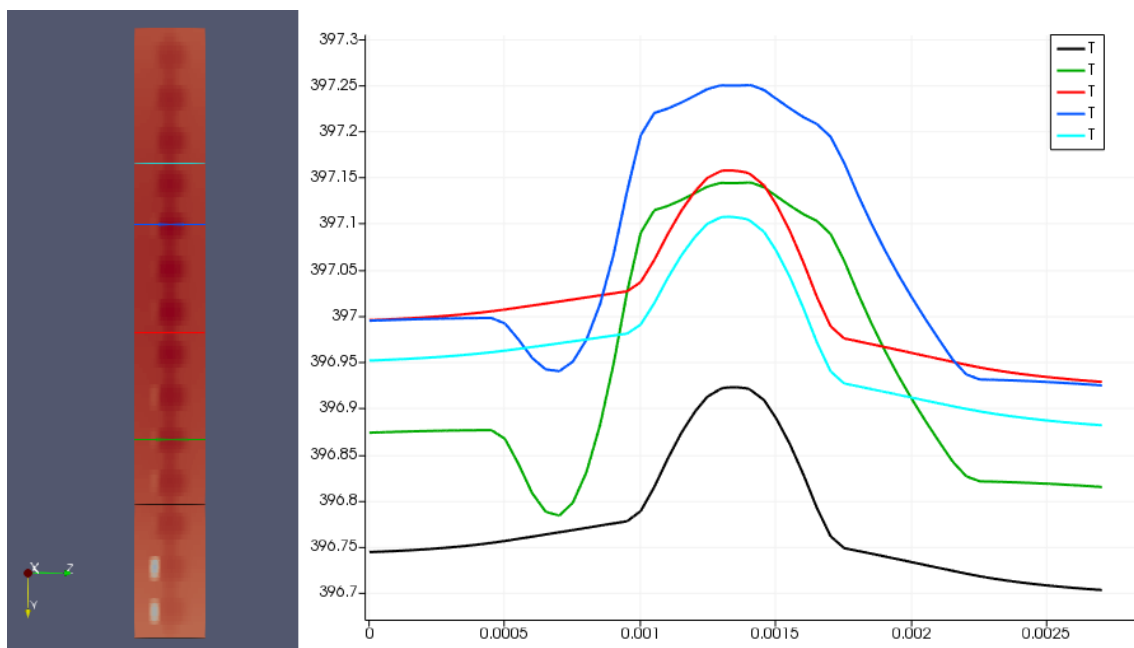


Figura 4.3: Grafica de evolución de temperaturas

En la gráfica de la Figura 4.3 vemos cómo evoluciona la temperatura en la sección central.

En el centro se aprecia una zona más caliente que corresponden a la zona de los electrodos, la capa catalítica y la membrana, que es donde se producen las reacciones electroquímicas y es la mayor fuente de calor, en las zonas que se encuentran al lado de los canales por los que circulan las especies (gráfica azul oscuro y verde) se ve como la zona con alta temperatura es mayor dado que al haber una mayor concentración de las especies se producen más reacciones ahí, en contraste con las zonas entre los canales en las que se alcanza menor temperatura.

También se aprecia como en los canales por los que circula el oxígeno (lado izquierdo de las gráficas azul oscuro y verde) se produce un descenso brusco de la temperatura mayor que en los canales por los que circula el hidrógeno (lado derecho de las gráficas azul oscuro y verde), esto se debe a que circula un caudal de oxígeno mayor que de hidrógeno.

Cabe destacar también como la temperatura aumenta a lo largo del recorrido de las especies por el calentamiento debido a las reacciones electroquímicas saliendo los productos a una temperatura cercana a su entorno (Figura 4.4 y Figura 4.5).

Por último, en la Figura 4.3, en los extremos se ve que tienen diferente temperatura, siendo menor en el lado derecho, que representa el lado superior de la pila, esto es debido a que la disipación de calor por convección natural es mayor en las caras que se encuentran hacia arriba.

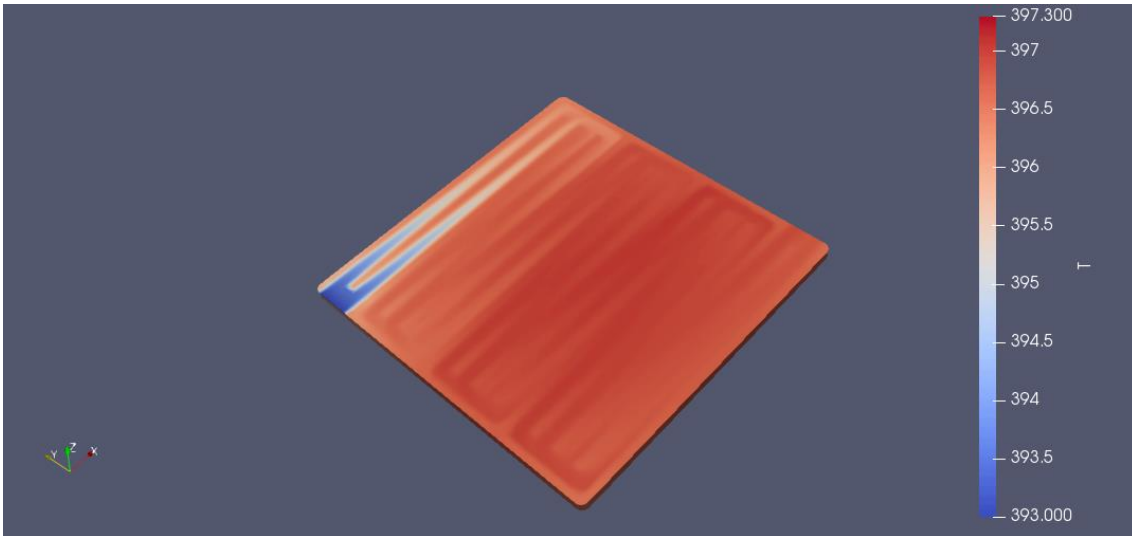


Figura 4.4: Distribución de temperatura en el canal de oxígeno

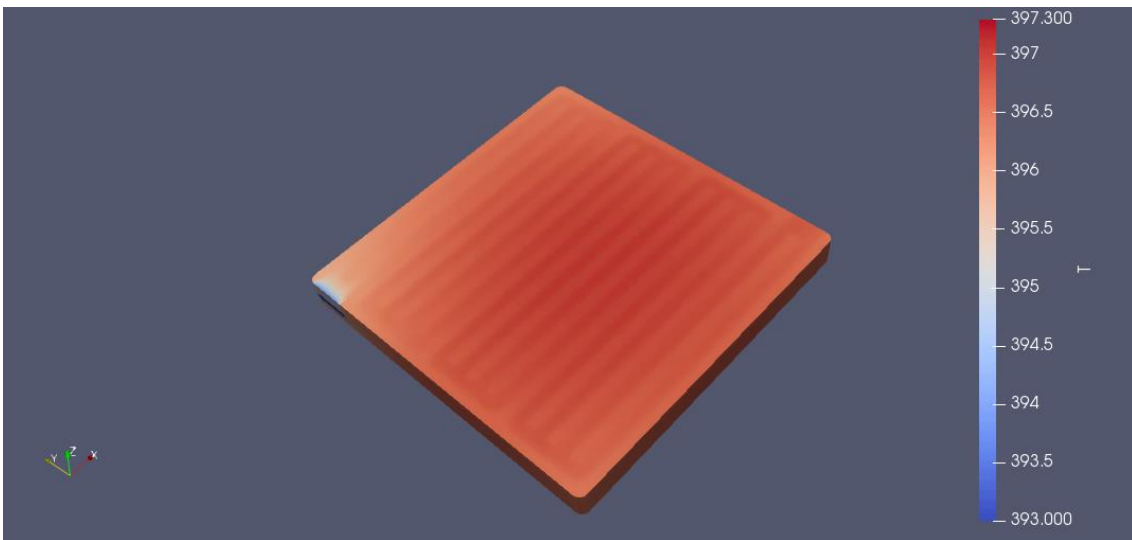


Figura 4.5: Distribución de temperatura en el canal de hidrogeno

Comparación de tiempos de simulación

Para ver el efecto que tiene la preiteración en la simulación vamos a comparar el tiempo que tarda en completarse la simulación con y sin preiteración. Para esto se realiza la simulación con varias densidades de corriente: 50, 200, 600, 1200, 2000, 3200, 4500, 5800, 6500, 7000, 7500, 7750 A/m².

La media de tiempo de ejecución de la preiteración es 0.17 segundos

En la Tabla 4.1 se compara el tiempo que simulación a diferentes intensidades:

Voltaje (V)	Intensidad (A/m ²)	Tiempo con preiteración TFG (s)	Tiempo sin preiteración TFG (s)	Mejora %	Temperatura final media (K)	Temperatura final preiteración (K)
1.01	50	4783	6077.77	-27.07	393.1043	393.0805
0.94	200	2329.97	4065.04	-74.47	393.4048	393.3268
0.87	600	1722.02	3483.68	-102.30	394.168	393.9666
0.81	1200	1628.33	2481.35	-52.39	395.2866	394.9257
0.75	2000	1455.67	2527.07	-73.60	396.7944	396.2425
0.68	3200	1486.29	2203.74	-48.27	399.1685	398.3479
0.59	4500	1666.94	2459.96	-47.57	401.9803	400.8717
0.50	5800	2270.2	2995.13	-31.93	405.1934	403.7648
0.43	6500	4290.2	5538.56	-29.10	407.2239	405.5688
0.37	7000	5445.25	7888.05	-44.86	408.9288	407.0465
0.28	7500	5747.96	7764.2	-35.08	411.1613	408.8365
0.20	7750	4722.92	6285.8	-33.09	412.9808	409.9864

Tabla 4.1: Comparación de resultados

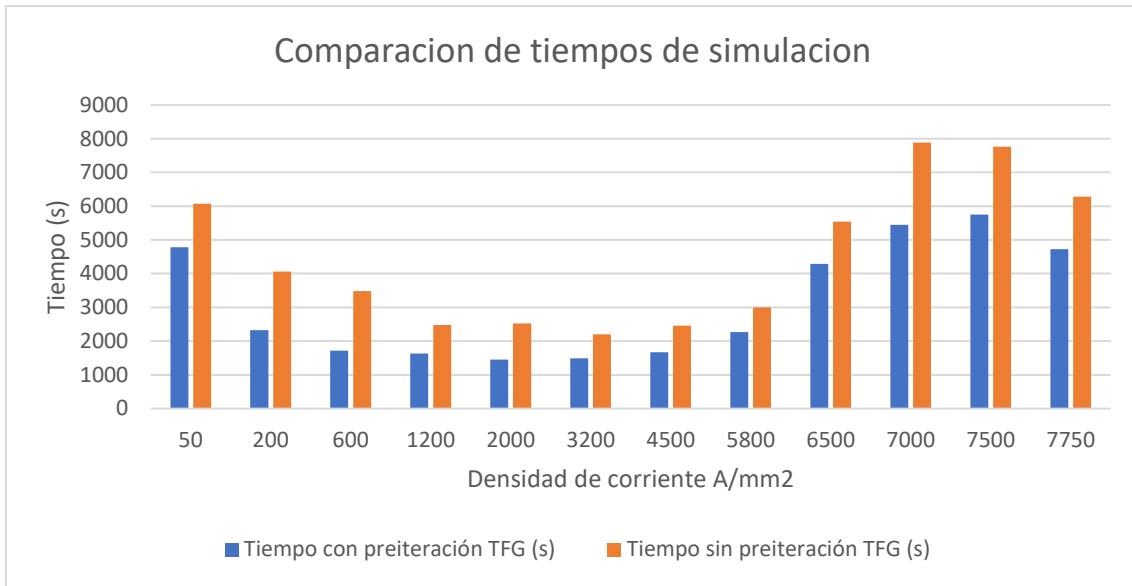


Figura 4.6: Comparación de tiempos de simulación

Como se aprecia en la Figura 4.6 con la preiteración se consigue una reducción significativa de los tiempos de simulación. Esta reducción del tiempo se produce porque la temperatura inicial para los cálculos sin la preiteración se considera 393 K y con la preiteración se calcula una temperatura más próxima a la temperatura final.

5 CONCLUSIONES.

A lo largo del trabajo se ha introducido que es una pila de combustible HTPEM y como produce corriente eléctrica, a través de la curva de polarización y se ha explicado brevemente los principales aspectos de OpenFOAM. Después se ha expuesto el modelo, proporcionado por el LIFTEC, que se usa para la realización de la simulación numérica.

Por otro lado, se explica el modelo de disipación de calor por convección natural que se añade como condición de contorno al modelo existente y el módulo preiteración que se han elaborado en este trabajo.

Por último, en los resultados finales vemos como la condición de contorno de convección natural nos proporciona unos valores de temperatura dentro de lo esperado, y comparando los tiempos de simulación del modelo sin la preiteración y con la preiteración vemos como se produce una reducción importante de este tiempo gracias a la preiteración que supone un cálculo rápido con poco coste computacional.

Vistos los resultados satisfactorios, en un futuro se podría incorporar a la preiteración ecuaciones que describan las interacciones entre celdas en un modelo de pila que incluya varias celdas, ya que al ser un modelo más complejo se beneficiara mucho de la reducción de tiempo de simulación que ofrece una preiteración así. En cuanto al modelo de convección, también sería aplicable a un modelo con múltiples celdas en el que permitiría calcular la disipación de calor en el conjunto de la pila o en celdas concretas dentro del modelo.

6 BIBLIOGRAFÍA

- Barbir, F. (2012). *PEM Fuel Cells: Theory and Practice*. Academic Press.
- Beale, S. B., Roth, H. K., Le, A., & Jeon, D. H. (2013). *Development of an open source software library for solid oxide fuel*.
- Bouaicha, A., Allagui, H., Aglizim, E.-H., Rouane, A., & Mami, A. (5 de Julio de 2016). Obtenido de <http://www.ijsei.com/papers/ijsei-55416-17.pdf>
- Cambridge, U. o. (2020). *Online Materials Science Learning*. Obtenido de <https://www.doitpoms.ac.uk/tlplib/fuel-cells/types.php>
- Çengel, Y. A. (2007). *Transferencia de Calor y Masa: Un enfoque practico*.
- Chevron Lubricants. (2015). Obtenido de Product Document information Center: <https://cglapps.chevron.com/msdspds/PDSDetailPage.aspx?docDataId=77158&docFormat=PDF>
- Holman, J. P. (2010). *Heat Transfer*.
- Li, S., Yuan, J., Xie, G., & Sundén, B. (2018). *Effects of agglomerate model parameters on transport characterization and performance of PEM fuel cells*.
- OpenCFD Limited. (2018). Obtenido de OpenFOAM User Guide: <https://www.openfoam.com/documentation/user-guide/>
- Velasco, A. R. (2017). *BOUNDARY CONDITIONS FOR CONVECTIVE HEAT TRANSFER IN OpenFOAM®*. Obtenido de https://foamingtime2.files.wordpress.com/2017/07/convection-bc_1.pdf
- Vilums, R. (2011). *Implementation of Transient Robin Boundary Conditions in OpenFOAM*. Obtenido de http://www.modlab.lv/docs/2011/OpenFOAM/Abstracts_R_Vilums_pdf.pdf

7 Anexos

7.1 Código de la condición de contorno.

```
/*-----*- C++ -*-----*\
|=====|
| \\ / Field | OpenFOAM: The Open Source CFD Toolbox |
| \\ / Operation | Version: 1.5 |
| \\ / And | Web: http://www.OpenFOAM.org |
| \\ \ M anipulation |
\*-----*/
FoamFile
{
    version 2.0;
    format ascii;
    class volScalarField;
    location "0";
    object T;
}
// ***** //
dimensions [0 0 0 1 0 0 0];
internalField uniform 393.1;
boundaryField
{
    interconnect_bottom
    {
        type groovyBC;
        variables ("Tinf=393;L=(0.023*0.023)/(4*0.023);g=9.81;Tfluido=(T+Tinf)/2;"
"betaF=0.000000004*pow(Tfluido,2)-0.000001448*Tfluido+0.000831505;")
    }
}
```

```

"rhoF=-0.00035*pow(Tfluido,2)-0.40994*Tfluido+1011.57359;"
"viscoCinemF=0.0000000005*pow(Tfluido,2)-0.0000004642*Tfluido+0.0001110221;"
"KF=-0.00014*Tfluido+ 0.16984;"
"cpF=-0.00320*pow(Tfluido,2)+6.85252*Tfluido+209.00043;"
"Ra= g*betaF*(T-Tinf)*pow(L,3)*rhoF*cpF/(viscoCinemF*KF);"
"NuB=(0.27*pow(Ra,0.25));"
"alpha=NuB*KF/L;";
valueExpression  "Tinf";
fractionExpression "1/(1+k/(alpha*mag(delta())))";
}
interconnect_top
{
    type      zeroGradient;
    type groovyBC;
variables ("Tinf=393;L=(0.023*0.023)/(4*0.023);g=9.81;Tfluido=(T+Tinf)/2;"
"betaF=0.000000004*pow(Tfluido,2)-0.000001448*Tfluido+0.000831505;"
"rhoF=-0.00035*pow(Tfluido,2)-0.40994*Tfluido+1011.57359;"
"viscoCinemF=0.0000000005*pow(Tfluido,2)-0.0000004642*Tfluido+0.0001110221;"
"KF=-0.00014*Tfluido+ 0.16984;"
"cpF=-0.00320*pow(Tfluido,2)+6.85252*Tfluido+209.00043;"
"Ra=g*betaF*(T-Tinf)*pow(L,3)*rhoF*cpF/(viscoCinemF*KF);"
"Nu=0.54*pow(Ra,0.25);"
"alpha=Nu*KF/L;";
valueExpression  "Tinf";
fractionExpression "1/(1+k/(alpha*mag(delta())))";
}
interconnectSides
{
    type groovyBC;

```

```

variables ("Tinf=393;espesor=0.0027;g=9.81;Tfluido=(T+Tinf)/2;"
"betaF=0.000000004*pow(Tfluido,2)-0.000001448*Tfluido+0.000831505;"
"rhoF=-0.00035*pow(Tfluido,2)-0.40994*Tfluido+1011.57359;"
"viscoCinemF=0.0000000005*pow(Tfluido,2)-0.0000004642*Tfluido+0.0001110221;"
"KF=-0.00014*Tfluido+ 0.16984;"
"cpF=-0.00320*pow(Tfluido,2)+6.85252*Tfluido+209.00043;"
"Ra=g*betaF*(T-Tinf)*pow(espesor,3)*rhoF*cpF/(viscoCinemF*KF);"
"Pr=viscoCinemF*rhoF*cpF/KF;"
"Nu=pow(0.825+(0.387*pow(Ra,0.1667))/pow(1+pow(0.5/Pr,0.5625),0.29629),2);"
"alpha=Nu*KF/espesor;");
valueExpression "Tinf";
fractionExpression "1/(1+k/(alpha*mag(delta())))";
}
airInlet
{
    type    fixedValue;
    value    393;
}
airOutlet
{
    type    zeroGradient;
}
airSides
{
    type    zeroGradient;
}
cathodeSides
{
    type groovyBC;
}

```

```

variables ("Tinf=393;espesor=0.0027;g=9.81;Tfluido=(T+Tinf)/2;"
"betaF=0.000000004*pow(Tfluido,2)-0.000001448*Tfluido+0.000831505;"
"rhoF=-0.00035*pow(Tfluido,2)-0.40994*Tfluido+1011.57359;"
"viscoCinemF=0.0000000005*pow(Tfluido,2)-0.0000004642*Tfluido+0.0001110221;"
"KF=-0.00014*Tfluido+ 0.16984;"
"cpF=-0.00320*pow(Tfluido,2)+6.85252*Tfluido+209.00043;"
"Ra=g*betaF*(T-Tinf)*pow(espesor,3)*rhoF*cpF/(viscoCinemF*KF);"
"Pr=viscoCinemF*rhoF*cpF/KF;"
"Nu=pow(0.825+(0.387*pow(Ra,0.1667))/pow(1+pow(0.5/Pr,0.5625),0.29629),2);"
"alpha=Nu*KF/espesor;");
valueExpression "Tinf";
fractionExpression "1/(1+k/(alpha*mag(delta())))";
}
cfISides
{
    type groovyBC;
variables ("Tinf=393;espesor=0.0027;g=9.81;Tfluido=(T+Tinf)/2;"
"betaF=0.000000004*pow(Tfluido,2)-0.000001448*Tfluido+0.000831505;"
"rhoF=-0.00035*pow(Tfluido,2)-0.40994*Tfluido+1011.57359;"
"viscoCinemF=0.0000000005*pow(Tfluido,2)-0.0000004642*Tfluido+0.0001110221;"
"KF=-0.00014*Tfluido+ 0.16984;"
"cpF=-0.00320*pow(Tfluido,2)+6.85252*Tfluido+209.00043;"
"Ra=g*betaF*(T-Tinf)*pow(espesor,3)*rhoF*cpF/(viscoCinemF*KF);"
"Pr=viscoCinemF*rhoF*cpF/KF;"
"Nu=pow(0.825+(0.387*pow(Ra,0.1667))/pow(1+pow(0.5/Pr,0.5625),0.29629),2);"
"alpha=Nu*KF/espesor;");
valueExpression "Tinf";
fractionExpression "1/(1+k/(alpha*mag(delta())))";
}

```

```

electrolyteSides
{
    type groovyBC;
variables ("Tinf=393;espesor=0.0027;g=9.81;Tfluido=(T+Tinf)/2;"
"betaF=0.000000004*pow(Tfluido,2)-0.000001448*Tfluido+0.000831505;"
"rhoF=-0.00035*pow(Tfluido,2)-0.40994*Tfluido+1011.57359;"
"viscoCinemF=0.0000000005*pow(Tfluido,2)-0.0000004642*Tfluido+0.0001110221;"
"KF=-0.00014*Tfluido+ 0.16984;"
"cpF=-0.00320*pow(Tfluido,2)+6.85252*Tfluido+209.00043;"
"Ra=g*betaF*(T-Tinf)*pow(espesor,3)*rhoF*cpF/(viscoCinemF*KF);"
"Pr=viscoCinemF*rhoF*cpF/KF;"
"Nu=pow(0.825+(0.387*pow(Ra,0.1667))/pow(1+pow(0.5/Pr,0.5625),0.29629),2);"
"alpha=Nu*KF/espesor;");
valueExpression "Tinf";
fractionExpression "1/(1+k/(alpha*mag(delta())))";
}
aflSides
{
    type groovyBC;
variables ("Tinf=393;espesor=0.0027;g=9.81;Tfluido=(T+Tinf)/2;"
"betaF=0.000000004*pow(Tfluido,2)-0.000001448*Tfluido+0.000831505;"
"rhoF=-0.00035*pow(Tfluido,2)-0.40994*Tfluido+1011.57359;"
"viscoCinemF=0.0000000005*pow(Tfluido,2)-0.0000004642*Tfluido+0.0001110221;"
"KF=-0.00014*Tfluido+ 0.16984;"
"cpF=-0.00320*pow(Tfluido,2)+6.85252*Tfluido+209.00043;"
"Ra=g*betaF*(T-Tinf)*pow(espesor,3)*rhoF*cpF/(viscoCinemF*KF);"
"Pr=viscoCinemF*rhoF*cpF/KF;"
"Nu=pow(0.825+(0.387*pow(Ra,0.1667))/pow(1+pow(0.5/Pr,0.5625),0.29629),2);"
"alpha=Nu*KF/espesor;");

```

```

valueExpression "Tinf";
fractionExpression "1/(1+k/(alpha*mag(delta())))";
}
anodeSides
{
    type groovyBC;
variables ("Tinf=393;espesor=0.0027;g=9.81;Tfluido=(T+Tinf)/2;"
"betaF=0.000000004*pow(Tfluido,2)-0.000001448*Tfluido+0.000831505;"
"rhoF=-0.00035*pow(Tfluido,2)-0.40994*Tfluido+1011.57359;"
"viscoCinemF=0.0000000005*pow(Tfluido,2)-0.0000004642*Tfluido+0.0001110221;"
"KF=-0.00014*Tfluido+ 0.16984;"
"cpF=-0.00320*pow(Tfluido,2)+6.85252*Tfluido+209.00043;"
"Ra=g*betaF*(T-Tinf)*pow(espesor,3)*rhoF*cpF/(viscoCinemF*KF);"
"Pr=viscoCinemF*rhoF*cpF/KF;"
"Nu=pow(0.825+(0.387*pow(Ra,0.1667))/pow(1+pow(0.5/Pr,0.5625),0.29629),2);"
"alpha=Nu*KF/espesor;");
valueExpression "Tinf";
fractionExpression "1/(1+k/(alpha*mag(delta())))";
}
fuelInlet
{
    type    fixedValue;
    value    393;
}
fuelOutlet
{
    type    zeroGradient;
}
fuelSides

```

```

{
  type    zeroGradient;
}
interconnect
{
  type    zeroGradient;
}
}
//
*****
** //

```

7.2 Código de la preiteración.

```

Info << "PreiterationTFG " << endl;
Info << "ExecutionTime = " << runTime.elapsedCpuTime() << " s\n\n" << endl;
scalar iH=ibar0.value();
scalar Tentrada = 393;
scalar Tcalculada;
scalar QinAno;
scalar QinCat;
List<scalar> rateInletCat;
rateInletCat.setSize(3);
List<scalar> rateInletAno;
rateInletAno.setSize(2);
forAll(airSpecies, I)
{
  rateInletCat[I] = mag(gSum
  (
    phiAir.boundaryFieldRef()[airInletID]*Yair[I].boundaryFieldRef()[airInletID]

```



```

));
scalar cpl = molarCpAir[I].polyVal(Tentrada)/(airSpecies[I].MW()*1e-3);
QinCat += rateInletCat[I]*cpl*Tentrada;
}
forAll(fuelSpecies, I)
{
    rateInletAno[I] = mag(gSum
    (
phiFuel.boundaryFieldRef()[fuelInletID]*Yfuel[I].boundaryFieldRef()[fuelInletID]
    ));
    scalar cpl = molarCpFuel[I].polyVal(Tentrada)/(fuelSpecies[I].MW()*1e-3);
    QinAno += rateInletAno[I]*cpl*Tentrada;
}
scalar Qin = (QinAno) + (QinCat);
Info << "Qin = " << Qin << endl;
scalar MW_O2 = airSpecies[0].MW();
scalar MW_H2O = airSpecies[2].MW();
scalar MW_H2 = fuelSpecies[0].MW();
scalar mO2cons = MW_O2 * iH / (FF*4) * cathodeArea ;
scalar mH2cons = MW_H2 * iH / (FF*2) * cathodeArea ;
scalar mH2Ogen = MW_H2O * iH / (FF*2) * cathodeArea ;
scalar mO2out = (rateInletCat[0])-mO2cons;
scalar mH2out = (rateInletAno[0])-mH2cons;
scalar mH2OoutCat = (rateInletCat[2])+mH2Ogen;
scalar Tout = 420;
scalar rel =0.5;
scalar Qtot;
scalar Qout ;
scalar loop1 = 0;

```

```

do{
Info << "LOOP = " << loop1 << endl;
Info << "QtotA = " << Qtot <<endl;
Info << "ToutA = " << Tout << endl;

scalar qO2out = mO2out * molarCpAir[0].polyVal(Tout)/(airSpecies[0].MW()*1e-3);
scalar qH2out = mH2out * molarCpFuel[0].polyVal(Tout)/(fuelSpecies[0].MW()*1e-3);

scalar qH2OoutCat =
mH2OoutCat*molarCpAir[2].polyVal(Tout)/(airSpecies[2].MW()*1e-3);

scalar qN2out=
rateInletCat[1] * molarCpAir[1].polyVal(Tout)/(airSpecies[1].MW()*1e-3);

scalar qH2Oano =
rateInletAno[1]*molarCpFuel[1].polyVal(Tout)/(fuelSpecies[1].MW()*1e-3);

Qout = (qO2out+qH2out+qH2OoutCat+qN2out+qH2Oano)*Tout ;
scalar yO2out = mO2out / (mO2out + mH2OoutCat + rateInletCat[1] ) ;
scalar yN2out = rateInletCat[1] / (mO2out + mH2OoutCat + rateInletCat[1] ) ;
scalar yH2OoutCat = mH2OoutCat / (mO2out + mH2OoutCat + rateInletCat[1] ) ;
scalar yH2out = mH2out / ( mH2out + rateInletAno[1] ) ;
scalar yH2OoutAno = rateInletAno[1] / ( mH2out + rateInletAno[1] ) ;
scalar yO2med = (gAverage(Yair[0].boundaryFieldRef()[airInletID])+ yO2out) /2;
scalar yN2med = (gAverage(Yair[1].boundaryFieldRef()[airInletID])+ yN2out)/2;
scalar yH2OmedCat =gAverage(Yair[2].boundaryFieldRef()[airInletID])+yH2OoutCat)/2;
scalar yH2med = (gAverage(Yfuel[0].boundaryFieldRef()[fuelInletID])+yH2out )/2;
scalar yH2OmedAno =
(gAverage(Yfuel[1].boundaryFieldRef()[fuelInletID]+ yH2OoutAno ))/2;

scalar xO2 =
(yO2med/airSpecies[0].MW())/(yO2med/airSpecies[0].MW()+yN2med/
airSpecies[1].MW()+yH2OmedCat/airSpecies[2].MW());

scalar xH2O =
(yH2OmedCat/airSpecies[2].MW())/(yO2med/airSpecies[0].MW()+
yN2med/airSpecies[1].MW()+yH2OmedCat/airSpecies[2].MW());

```

```

scalar xH2 =
(yH2med/fuelSpecies[0].MW())/(yH2med/fuelSpecies[0].MW()+
yH2OmedAno/fuelSpecies[1].MW());

scalar A = -alphaAnodeElectrolyte.value()*F/(Rgas*Tout);
scalar B = (2-alphaAnodeElectrolyte.value()*F/(Rgas*Tout);

scalar iOAnodep =

gammaAnodeElectrolyte.value()*Foam::exp(-1.*EAnodeElectrolyte.value()/
(Rgas*Tout)*(1-Tout/Tri0))*Foam::pow((pAtm*xH2/pAtm), aElectrolyte.value());

testFunctionAnode tfA(iH, iOAnodep, A , B);

scalar etaAnode = RiddersRoot<testFunctionAnode>(tfA, 1e-10).root(0., 1.2,1);

scalar Hi = 4.04e-5;

A = alphaCathodeElectrolyte.value()*F/(Rgas*Tout);
B = -(1-alphaCathodeElectrolyte.value()*F/(Rgas*Tout);

scalar M = 1-epsilonCL;

scalar Z = (ragg.value() + delta.value())*delta.value()/(ragg.value()*agg*Do.value());

scalar X = ragg.value()/(3*Foam::sqrt(Foam::pow(Liagg.value(),1.5) *Do.value()));

scalar Y =

gammaCathodeElectrolyte.value()*Foam::exp(-1.*ECathodeElectrolyte.value()/
(Rgas*Tout)*(1-Tout/Tri0))/(4*F*M*3.39);

scalar N = 4 * F * pAtm*xO2*Hi*(hE.value()/12);

testFunction2 tf2(iH, A , B , X , Y , Z , N , M);

scalar etaCathodeM=
RiddersRoot<testFunction2>(tf2, 1e-10).root(0.01,1.2,V.value());

scalar DS;

scalar Qrxn =1;

forAll(airSpecies, i)
{
    if (airSpecies[i].rSign() == -1)

```

```

{
  scalar stoiCoeffl = rxnSpCoef[airSpecies[i].name()];
  Qrxn *= Foam::pow(xO2, stoiCoeffl);
  scalar Si = stoiCoeffl *
  (
    airSpecies[i].sForm()
    + molarCpAir[i].polyIntS(Tr, Tout)
  );
  DS += Si;
}
else if (airSpecies[i].rSign() == 1)
{
  scalar stoiCoeffl = rxnSpCoef[airSpecies[i].name()];
  Qrxn /= Foam::pow(xH2O, stoiCoeffl);
  scalar Si = stoiCoeffl *
  (
    airSpecies[i].sForm()
    + molarCpAir[i].polyIntS(Tr, Tout)
  );
  DS -= Si;
}
DS += Rgas * Foam::log(Qrxn);
}
forAll(fuelSpecies, i)
{
  scalar stoiCoeffl = rxnSpCoef[fuelSpecies[i].name()];
  if (fuelSpecies[i].rSign() == -1)
  {
    Qrxn *= Foam::pow(xH2, stoiCoeffl);
  }
}

```

```

    scalar Si = stoiCoeffl *
    (
        fuelSpecies[i].sForm()
        + molarCpFuel[i].polyIntS(Tr, Tout)
    );

    DS += Si;
}
DS += Rgas * Foam::log(Qrxn);
}
scalar electrolyteVolume = Foam::gSum
(
    electrolyteMesh.V()
);
scalar Qele =
Foam::pow(iH,2)*electrolyteVolume/Kprot.value()+iH*(etaAnode+etaCathodeM+
Tout*DS/(2*FF))*cathodeArea;
scalar Tfluido = (Tout+Tentrada)/2;
Info << "Tfluido = " << Tfluido << endl;
scalar betaF =
0.000000004*Foam::pow(Tfluido,2)-0.000001448*Tfluido+0.000831505;
scalar rhoF = -0.00035*Foam::pow(Tfluido,2)-0.40994*Tfluido+1011.57359;
scalar viscoCinemF=
0.0000000005*Foam::pow(Tfluido,2)-0.0000004642*Tfluido+0.0001110221;
scalar KF= -0.00014*Tfluido+0.16984;
scalar cpF=-0.00320*Foam::pow(Tfluido,2)+6.85252*Tfluido+209.00043;
scalar celdas = 1;
scalar cellVolume = Foam::gSum(mesh.V());
scalar ancho = Foam::sqrt(cathodeArea);
scalar espesor = cellVolume/cathodeArea;

```

```

scalar lados = 4;

scalar Lvert = espesor*celdas;

scalar LHoriz = Foam::pow(ancho,2)/(4*ancho);

scalar g = 9.81;

scalar Pr =viscoCinemF*rhoF*cpF/KF;

scalar AconvV = espesor*ancho*lados;

scalar AconvH = Foam::pow(ancho,2);

scalar RaV =

g*betaF*(Tout-Tentrada)*Foam::pow(Lvert,3)*rhoF*cpF/(viscoCinemF*KF);

scalar NuV = Foam::pow(0.825+(0.387*Foam::pow( RaV ,0.1667)))/Foam::pow(1+
Foam::pow(0.5/Pr,0.5625),0.29629),2);

scalar hV = KF*NuV/(Lvert);

Info<< "NuV = " <<NuV << "; hV = " << hV << endl;

scalar QconvV = hV *AconvV *(Tout-Tentrada);

scalar RaH =

g*betaF*(Tout-Tentrada)*Foam::pow(LHoriz,3)*rhoF*cpF/(viscoCinemF*KF);

scalar NuTop =0.54*Foam::pow(RaH,0.25);

scalar NuBot=0.27*Foam::pow(RaH,0.25);

scalar hTop = KF*NuTop/LHoriz;

scalar hBot = KF*NuBot/LHoriz;

scalar QconvTop = hTop * AconvH *(Tout-Tentrada);

scalar QconvBot = hBot * AconvH *(Tout-Tentrada);

scalar Qconv = QconvV + QconvTop + QconvBot ;

Tcalculada=(Qin+Qele+(hV*AconvV+hTop*AconvH+hBot*AconvH)*Tentrada)/
((qO2out+qH2out+qH2OoutCat+qN2out+qH2Oano)+(hV*AconvV+hTop*AconvH+
hBot*AconvH));

Info << "betaF = " << betaF << "; rhoF = " << rhoF << "; viscoCinemF = " <<
viscoCinemF << "; KF = " << KF << "; cpF = " << cpF << endl;

Qtot = (Qin) - (Qout) - Qconv + Qele ;

Info << "Tcalculada = " << Tcalculada << endl;

```

```

Tout += (Tcalculada - Tout) * rel;
Info << "Tout = " << Tout << endl;
Info << "Qtot = " << Qtot << endl;
Info << "Qout = " << Qout << endl;
Info << "Qconv = " << Qconv << endl;
Info << "+QconvV = " << QconvV << endl;
Info << "+QconvTop = " << QconvTop << endl;
Info << "+QconvBot = " << QconvBot << endl;
Info << "Qele = " << Qele << endl;
loop1++;
} while (mag(Qtot)>0.001);
Info << "FIN BUCLE " << endl;
Info << "ToutFIN = " << Tout << endl;
Info << "QtotFIN = " << Qtot << endl;
Tcell.primitiveFieldRef() =Tout;
TcellOLD.primitiveFieldRef() =Tout;
TcellOLD.correctBoundaryConditions();
Info << "Tcalculated = " << Tout << endl;
Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s\n\n" << endl;
# include "mapFromCell.H"

```