



Calibración y validación para un caso de flujo turbulento resuelto a través del módulo de ecuaciones RANS SST en TAU-Code

Augusto Manuel Dovetta

Universidad Nacional de Rosario
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Escuela de Ingeniería Mecánica
Rosario, Argentina
2023

Calibración y validación para un caso de flujo turbulento resuelto a través del módulo de ecuaciones RANS SST en TAU-Code

Augusto Manuel Dovetta

Proyecto Final de Ingeniería Mecánica presentado como requisito parcial para optar al
título de:

Ingeniero/a Mecánico/a

Director/a:

Ing. Pedro Sismondi – Ing. Guillermo Rodríguez – Ing. Juan Sismondi

Universidad Nacional de Rosario
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Escuela de Ingeniería Mecánica
Rosario, Argentina
2023

*Dedicado a Rocco, mi mejor amigo. Te
extraño todos los días compañero.*

Agradecimientos

A mi familia por estar siempre presente en este camino. Sin dudas alguna, sin su apoyo el desenlace no hubiera sido mismo.

A la Facultad de Ciencias Exactas, Ingeniería y Agrimensura (FCEIA) de la Universidad Nacional de Rosario (UNR) por haberme formado como persona y profesional.

A los docentes, compañeros y personal no docente de la Escuela de Ingeniería Mecánica de la FCEIA por los momentos compartidos, la educación recibida y las instalaciones prestadas.

Al Ministerio de Educación de la Republica Argentina y al Servicio de Intercambio Académico Alemán (DAAD) por haberme dado la posibilidad de realizar esta etapa de mis estudios en Alemania.

Al Instituto de Mecánica de Fluidos de la Universidad Técnica de Braunschweig por abrirme sus puertas y presentarme el trabajo realizado.

A mi tutor M. Sc. Gokul Subbian por su tiempo, apoyo, consejo y supervisión en este proyecto.

A la catedra de Proyecto Final de Ingeniería Mecánica por el tiempo brindado y la gran predisposición mostrada.

A mi grupo de amigos y compañeros de estudio por los momentos compartidos y el constante apoyo.

A toda persona que haya participado en mi proceso de formación como Ingeniero.

Resumen

El presente proyecto tiene como objetivo principal la calibración de un caso de validación para simulaciones RANS SST en el programa TAU-Code. El caso en estudio corresponde al ejemplo clásico conocido como “2D Backward Facing Step” brindado por la NASA.

Mediante esta calibración, se busca comprobar el correcto funcionamiento del programa, para así respaldar futuras simulaciones y establecer un marco de comparación para los posteriores estudios en el Instituto de Mecánica de Fluidos de la Universidad Técnica de Braunschweig.

A través de la aplicación de técnicas de optimizaciones bayesianas, se logró llevar a cabo exitosamente la calibración. Para este propósito, se utilizó la herramienta computacional UQLab, la cual demostró ser altamente efectiva en este contexto. Además de los resultados obtenidos, se busca crear un caso ejemplo de uso del software para abordar problemas similares.

Se realizaron simulaciones adicionales que corroboraron y validaron los resultados obtenidos, constituyendo un respaldo sólido para la calibración realizada. De tal manera, este proyecto demuestra el desempeño adecuado del módulo de simulaciones RANS SST del software TAU-Code, brindando así seguridad y confianza en la utilización del mismo a largo plazo.

Palabras clave: Bayesian optimization – TAU-Code – RANS SST – UQLab – 2DBFS

Abstract

The main objective of this project is the calibration of a validation case for RANS SST simulations in TAU-Code. The case under study corresponds to the classic example known as "2D Backward Facing Step" provided by NASA.

The aim of this calibration is to check the correct functioning of the program in order to support future simulations and to establish a framework of comparison for subsequent studies at the Institute of Fluid Mechanics of the Technical University of Braunschweig.

Through the application of Bayesian optimisation techniques, the calibration was successfully carried out. For this purpose, the computational tool UQLab was used, which proved to be highly effective in this context. In addition to the results obtained, the aim is to create an example case of the use of the software to address similar problems.

Additional simulations were carried out to corroborate and validate the results obtained, constituting a solid support for the calibration performed. In this way, this project demonstrates the adequate performance of the RANS SST simulation module of the TAU-Code software, thus providing security and confidence in its long-term use.

Keywords: Bayesian optimization – TAU-Code – RANS SST – UQLab – 2DBFS

Contenido

Agradecimientos	4
Resumen	5
Abstract	6
Contenido	7
Introducción	8
Información preliminar y marco teórico	14
Datos iniciales	14
Método Propuesto – Inferencia Bayesiana	18
Modelo subrogado – PCE	24
Calibración	27
Herramienta computacional – UQLab	27
Desarrollo del código	29
Carga de datos	30
Pasos de calibración	33
Postproceso de resultados	37
Validación	42
DLR TAU-Code	42
Modelo CAD y mallado	43
Pasos de simulación	44
Primera simulación	45
Segunda simulación	50
Simulación final calibrada	56
Conclusiones y recomendaciones	63
Conclusiones	63
Recomendaciones	64
Bibliografía	65
Índice de figuras:	67
Anexo 1: Código MatLab y Uqlab	69
Anexo 2: Código TAU-Code	76

Introducción

La mecánica de fluidos computacional (CFD por sus siglas en inglés) ha adquirido una gran utilidad y relevancia a medida que la tecnología avanza y las necesidades de la **industria y la ciencia** requieren **soluciones eficientes y confiables** en el estudio de flujos complejos. A lo largo del tiempo, diversos enfoques y modelos han surgido para abordar las necesidades y desafíos de este campo en constante evolución.

El comportamiento de un fluido se encuentra gobernado por las ecuaciones diferenciales parciales de **Navier-Stokes**, las cuales describen la conservación de masa, momento y energía en función del tiempo y la posición. Si bien estas ecuaciones proporcionan la **solución precisa**, su alta **complejidad matemática y computacional** hace **inviable su utilización** en la mayoría de los casos, lo que lleva a emplear modelos simplificados en su lugar.

Aquí es donde los **modelos RANS** entran en juego. Estos modelos se basan en la idea de **promediar las ecuaciones de Navier-Stokes en el tiempo** para obtener valores medios, lo cual permite **simplificar en gran medida los cálculos**. En lugar de resolver las ecuaciones para cada partícula y en cada instante, los modelos RANS otorgan resultados promediados en el tiempo, que aunque son **soluciones simplificadas**, **representan adecuadamente los fenómenos físicos**, especialmente en casos que involucren turbulencia [14].

Dentro de los modelos RANS, el modelo **RANS SST** (Shear Stress Transport) destaca particularmente por su capacidad para proporcionar una **predicción precisa del flujo turbulento**. Este combina ventajas de modelos más simples, como lo son los modelos $k-\omega$ y $k-\epsilon^1$, logrando así una descripción más detallada de la turbulencia [15].

En el Instituto de Mecánica de Fluidos (Institut für Strömungsmechanik) de la Universidad Técnica de Braunschweig (Technische Universität Braunschweig), se cuenta con un

¹ Los modelos $k-\omega$ y $k-\epsilon$ son dos de los modelos dentro de la familia RANS más utilizados para la simulación y análisis de la turbulencia en la mecánica de fluidos computacional. Mas información sobre los mismos disponible en: <https://turbmodels.larc.nasa.gov/wilcox.html> - <https://turbmodels.larc.nasa.gov/ke-chien.html>

software de CFD de desarrollo propio llamado **DLR TAU-Code**. El mismo es utilizado en los diversos estudios y proyectos que el Instituto lleva a cabo.

Una práctica muy común en el marco científico es buscar la reproducibilidad de resultados de bibliografía previamente comprobada. Dentro de la base de datos que la NASA (National Aeronautics and Space Administration) provee libremente a la comunidad, se puede encontrar el siguiente estudio llamado **2DBFS: 2D Backward Facing Step**, el cual propone un caso de validación para diversos modelos de cálculo de flujo turbulento.

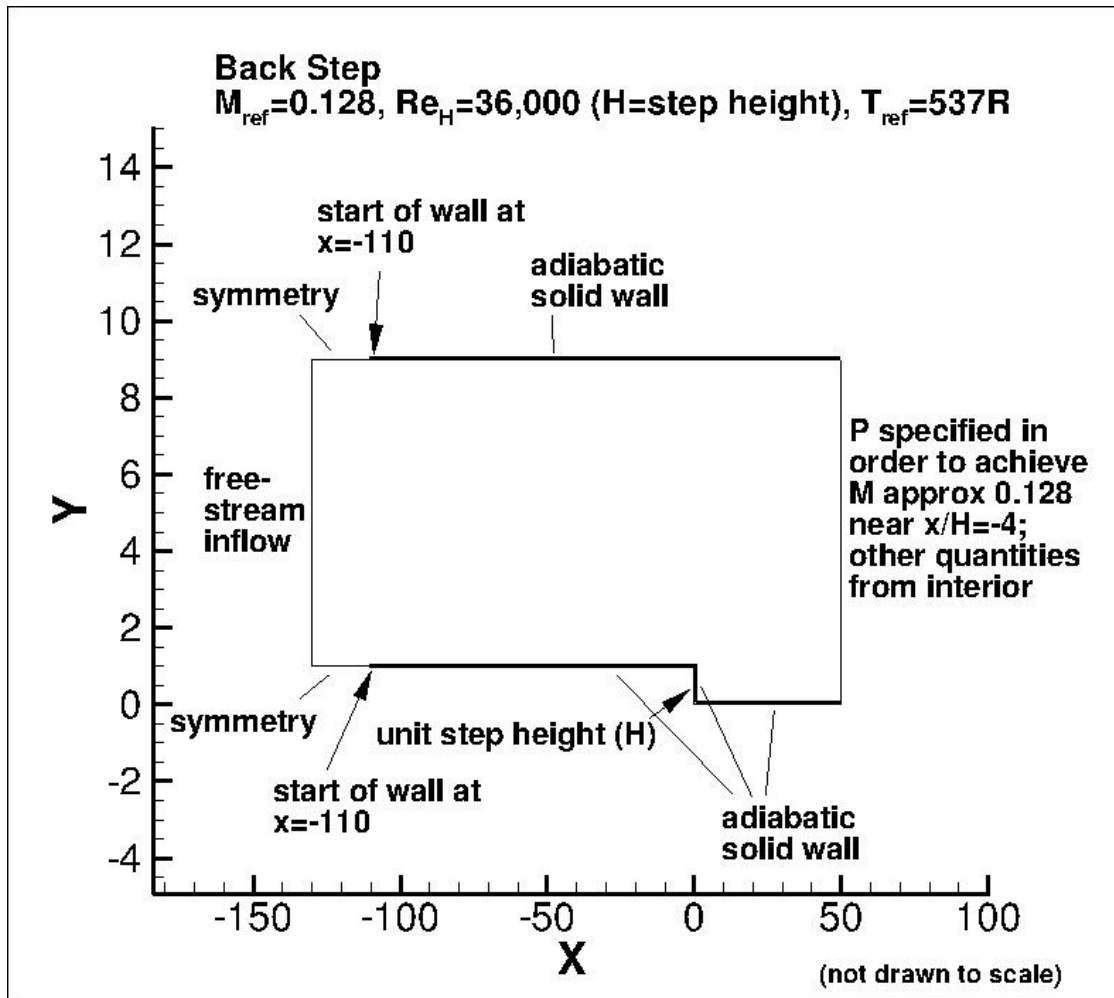


Figura 1-1: 2DBFS: 2D Backward Facing Step, Ilustración del problema [13]:

Desde el Instituto de Mecánica de Fluidos de la Universidad Técnica de Braunschweig, se pretende obtener la reproducibilidad de resultados para dicho problema utilizando como modelo de flujo turbulento **STT** siendo este resuelto a través del software propio **TAU-Code**. Esto permitirá realizar una **validación de la confiabilidad y precisión** del

software al trabajar utilizando el modelo STT, ya que, si se lograra reproducir los resultados de un análisis altamente comprobado, evidenciará el **correcto funcionamiento** del mismo y **respaldará futuros estudios y resultados**.

En cuanto a las condiciones presentes en el caso en estudio, desde la misma web de la NASA, se aclaran lo siguiente:

En este caso, una capa límite turbulenta se encuentra con un retroceso repentino que provoca la separación del flujo. A continuación, el flujo vuelve a unirse y se recupera aguas abajo del escalón. El número de Reynolds basado en el espesor del momento de la capa límite antes del escalón es 5000. Esto corresponde a un número de Reynolds de aproximadamente 36.000 basado en la altura del escalón H . El espesor de la capa límite antes del escalón es de aproximadamente $1,5H$.

Las condiciones límite apropiadas para el CFD se muestran en la siguiente figura. Aparte de una corta región con simetría impuesta (para evitar posibles incompatibilidades entre el flujo de entrada libre y la CB de la pared), tanto la pared inferior como la superior se tratan como paredes viscosas.

En este caso, la longitud de entrada antes de la zona de interés (cerca de $x=0$) se ha ajustado para que la capa límite turbulenta que se desarrolla de forma natural en la pared inferior en la solución CFD crezca hasta aproximadamente el grosor correcto y produzca aproximadamente el coeficiente correcto de fricción de la piel de la pared antes del escalón. La contrapresión se ajusta para obtener aproximadamente el número de Mach correcto ($M=0,128$) antes del escalón.

El flujo en BFS es altamente tridimensional y presenta una amplia gama de fenómenos físicos, incluyendo la separación del flujo, recirculación, formación de vórtices, transferencia de calor y transferencia de masa. Siendo así un caso base ideal para el estudio a llevar a cabo, ya que presenta una combinación perfecta entre sencillez geométrica y complejidad en los fenómenos físicos involucrados de la cual se cuenta con resultados previos respaldados. [13]

El modelo de turbulencia SST es dependiente de cinco **parámetros adimensionales** llamados ($C\omega 4 - cr1 - cr2 - cr3 - \zeta 2$), los cuales cumplen el rol de **constantes de corrección del modelo**. Estos parámetros **permiten ajustar el modelo matemático a la realidad**. Solo en base a experiencias previas se conocen intervalos numéricos que acotan los valores que dichos parámetros pueden tomar. Resumidamente, cada uno de ellos cumple una función específica en el modelo [15]:

- **$C\omega 4$** : Acompaña a la tasa de disipación de energía turbulenta ω y ajusta cómo se

modela la disipación de energía turbulenta en el flujo.

- **cr1**: Influye en la tasa de producción de energía cinética turbulenta k en la ecuación del modelo $k - \omega$, controlando cómo se estima la generación de turbulencia en función de las tensiones de corte en el flujo.
- **cr2**: También proviene del modelo $k - \omega$, y afecta la disipación de energía cinética turbulenta k . Modela la pérdida de energía cinética turbulenta debido a la disipación viscosa.
- **cr3**: Describe la viscosidad turbulenta μ_t , influyendo en cómo se modela la viscosidad turbulenta en función de la disipación de energía turbulenta ω y la escala de turbulencia.
- **ζ2**: Proviene del modelo $k - \varepsilon$ y afecta cómo se estima la tasa de disipación de energía cinética turbulenta ε , controlando cómo se modela la disipación de energía cinética turbulenta en el flujo.

El objetivo principal de este informe no es profundizar en las ecuaciones que gobiernan este fenómeno, sino más bien **enfocarse en la calibración del método**. Mas precisamente, la tarea a llevar a cabo será **determinar los cinco valores de dichos parámetros que mejor ajusten los resultados de la simulación numérica a la realidad**. En otras palabras, se busca encontrar los coeficientes de corrección más adecuados para mejorar la precisión del modelo en la simulación.

Para el interés del lector y si el mismo desea profundizar en el modelo matemático, se deja adjunto el siguiente enlace, el cual presenta el desarrollo detallado del modelo RANS SST: <https://turbmodels.larc.nasa.gov/sst.html#sst-std>

En cuanto a los resultados que el modelo otorga, se distinguen los listados a continuación:

- Coeficiente presión sobre la pared inferior del canal.
- Coeficiente de fricción sobre la pared inferior del canal.
- Perfil de velocidad para distintas posiciones.
- Tensiones de Reynolds para distintas posiciones.

Es posible observar que en todas las gráficas presentes en la figura 1-2, los resultados se encuentran expresados en función de **x/H e y/H** . El propósito de esto, es que los **resultados sean adimensionales**, siendo **H la altura del escalón**, la posición X e Y , se definen en relación a dicho valor. Lo mismo ocurre con los valores de velocidad, siendo

Uref el valor de simulación para velocidad en la dirección X, los resultados se encuentran expresados de forma adimensional en referencia al valor utilizado en la simulación.

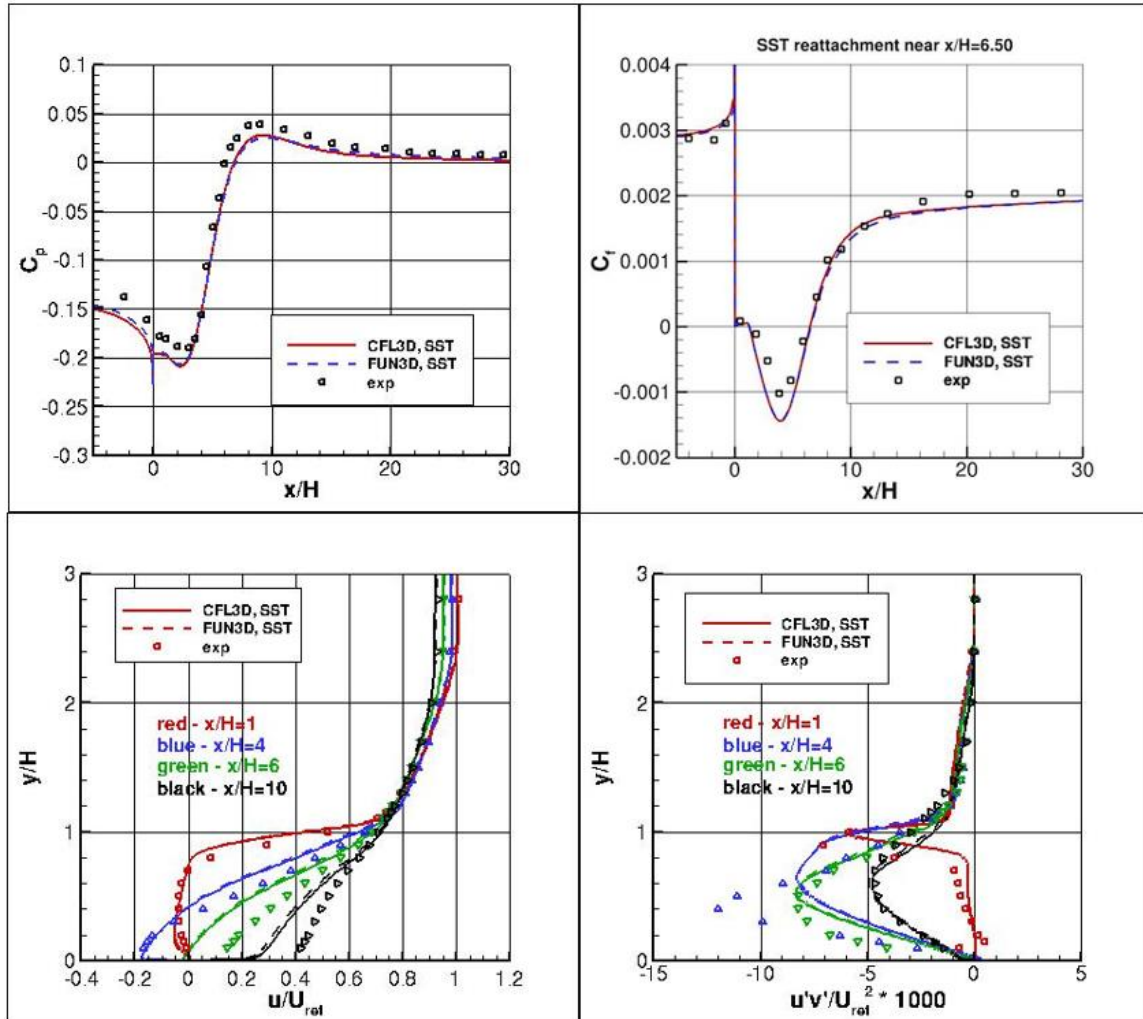


Figura 1-2: Resultados experimentales NASA [13]:

Al no tener forma de estimar los valores de los parámetros de corrección del modelo y al ser estos cinco parámetros dependientes entre sí, no es posible en términos prácticos realizar una calibración manual. Ya que cualquier modificación en uno de estos, modifica constantemente los resultados, siendo prácticamente imposible llegar a conclusiones de esta manera. Por lo tanto, es aquí donde se presenta una de las mayores dificultades al trabajar con el modelo de turbulencia anteriormente nombrado y donde radicara el desarrollo próximo en el presente informe.

Se esperan llegar a **resultados semejantes** a los obtenidos por la NASA en sus simulaciones en tanto a valores de **perfiles de velocidad, coeficientes de presión y rozamiento**. Se conoce de antemano que el programa **TAU-Code no otorga resultados correctos en cuanto a tensiones de Reynolds**. Sin embargo, se incluyen dichos resultados dentro de la calibración, para así poder tener una referencia del error del mismo.

El estudio completo consistirá principalmente en **dos etapas** bien definidas. Inicialmente, será necesario definir una **estrategia de calibración** de parámetros y obtener una o varias combinaciones que puedan cumplir el objetivo. Mientras que, en segundo lugar, será necesario **validar los resultados**, llevando a cabo la simulación con los nuevos parámetros obtenidos.

Información preliminar y marco teórico

En el presente capítulo se hará de una descripción del estado inicial en el cual se presenta este proyecto para su calibración. Además, se introducirá el método de calibración utilizado, justificando los aspectos teóricos que respaldan al mismo como el indicado para el estudio en cuestión.

Datos iniciales

Al no tener manera de poder determinar ($C\omega_4 - cr_1 - cr_2 - cr_3 - \zeta_2$) de primera mano, experiencia y conocimientos previos en este tipo de trabajos, sugieren y recomiendan realizar un **análisis en base de probabilidad y estadística** entre los resultados otorgados por diversas soluciones y los resultados reales que se quieren obtener (presentes previamente en la figura 1-2).

Por razones que serán detalladas la siguiente sección, inicialmente se determinaron **trescientas combinaciones aleatorias** de los coeficientes ($C\omega_4 - cr_1 - cr_2 - cr_3 - \zeta_2$) y se llevan a cabo las **trescientas simulaciones numéricas**, obteniendo así una gran base de datos que servirá de referencia para los siguientes pasos de este proyecto. Vale aclarar que mientras con más información de este tipo se cuente, los resultados futuros serán más precisos.

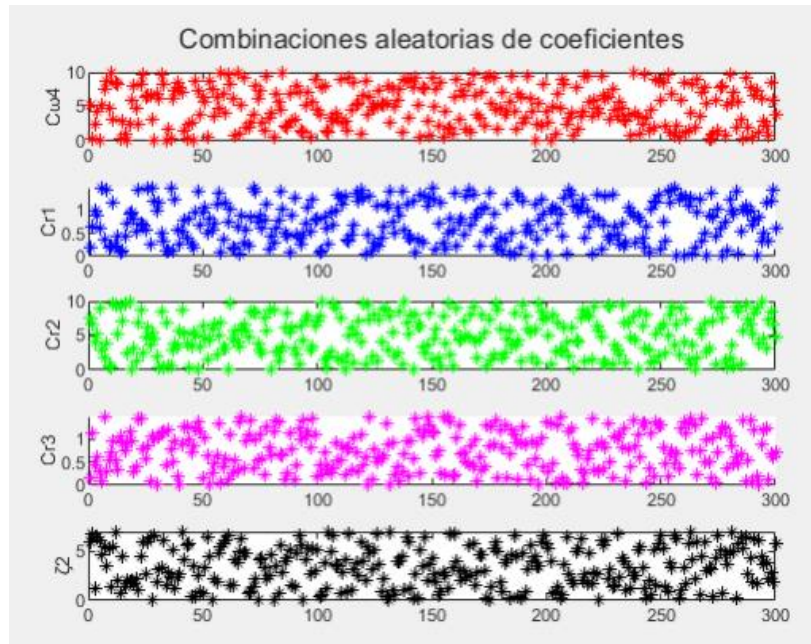


Figura 2-1: Combinación de parámetros en orden de uso:

La aleatoriedad en la selección también se puede comprobar y notar en la figura 2-2 donde se ordenan los parámetros en orden ascendente. Al ordenarse linealmente entre el mínimo y el mayor, se comprueba que los mismos se encuentran distribuidos uniformemente.

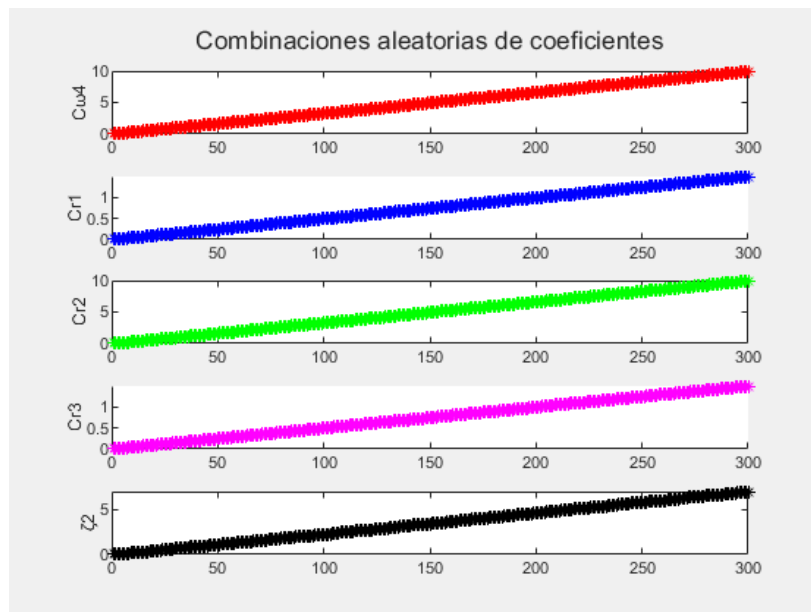


Figura 2-2: Combinación de parámetros en orden ascendente:

De tal manera, se muestran a continuación los treientos resultados combinados en una sola grafica para cada variable de interés. Se puede observar claramente que los resultados numéricos obtenidos respetan similitudes entre sí en cuanto a la forma de su distribución, a la vez coincidente con la información real de la figura 1-2 pero lógicamente distribuidos entre un rango de valores que justamente es lo que se desea calibrar.

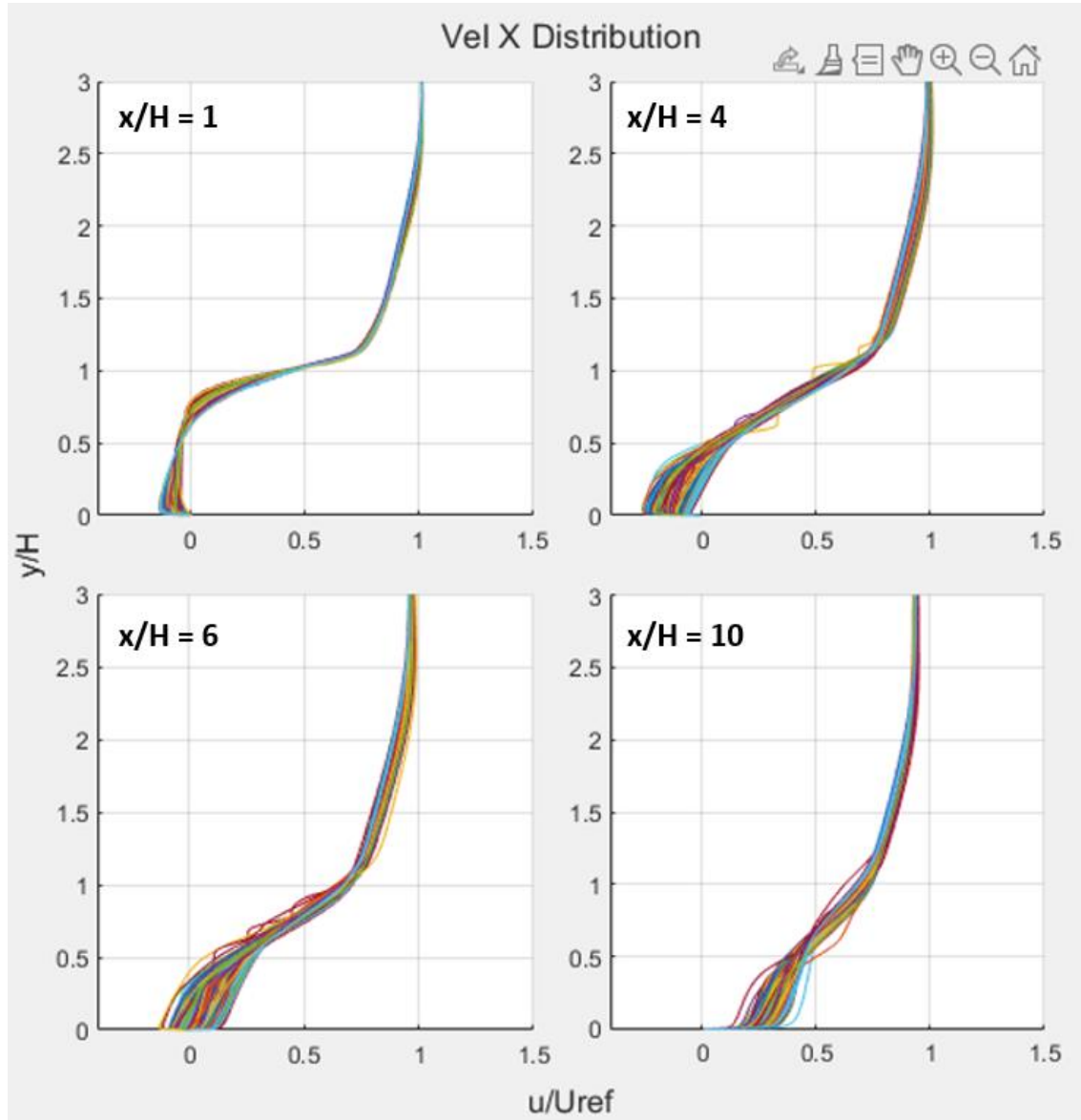


Figura 2-3: Distribución inicial de perfiles de velocidad:

En una primera instancia, se encuentran en la figura 2-3 los resultados correspondientes a la distribución de velocidad en la dirección x/H , para las diferentes alturas en posiciones

de interés y representativas tal y como $x=1$; 4; 6 y 10.

De la misma manera que detallada anteriormente, se otorgan en la figura 2-4, los resultados de coeficientes de presión y fricción sobre la pared inferior del problema

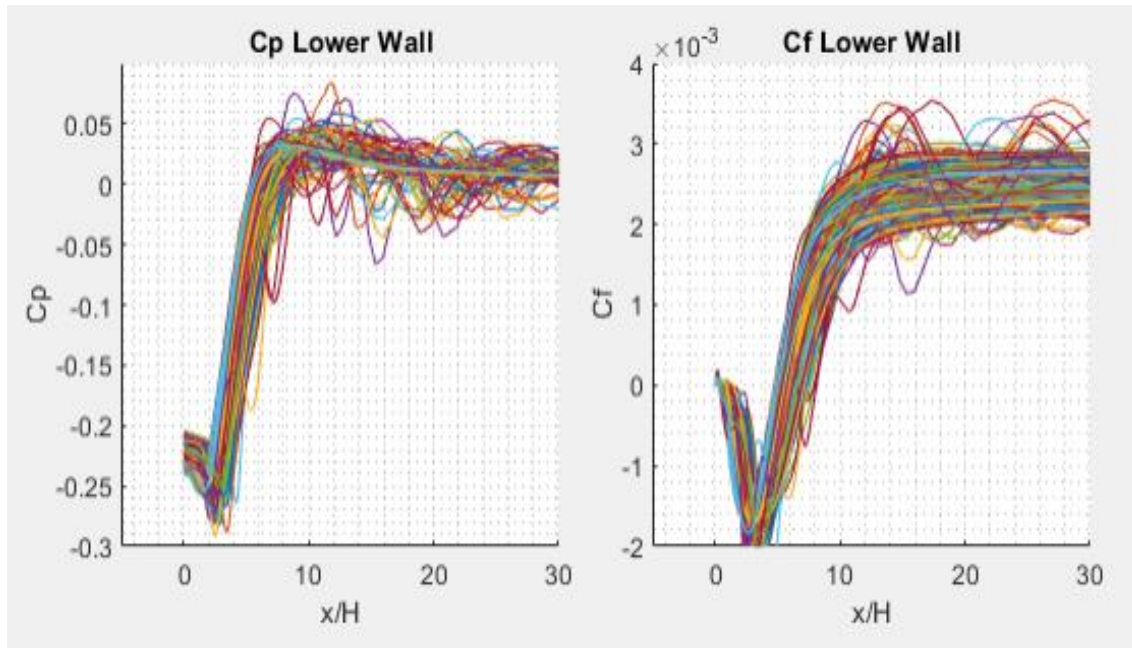


Figura 2-4: Distribución inicial de C_p y C_f :

Las distribuciones de resultados de tensiones de Reynolds son mostradas en la figura 2-5 para las mismas posiciones que se obtuvo la velocidad.

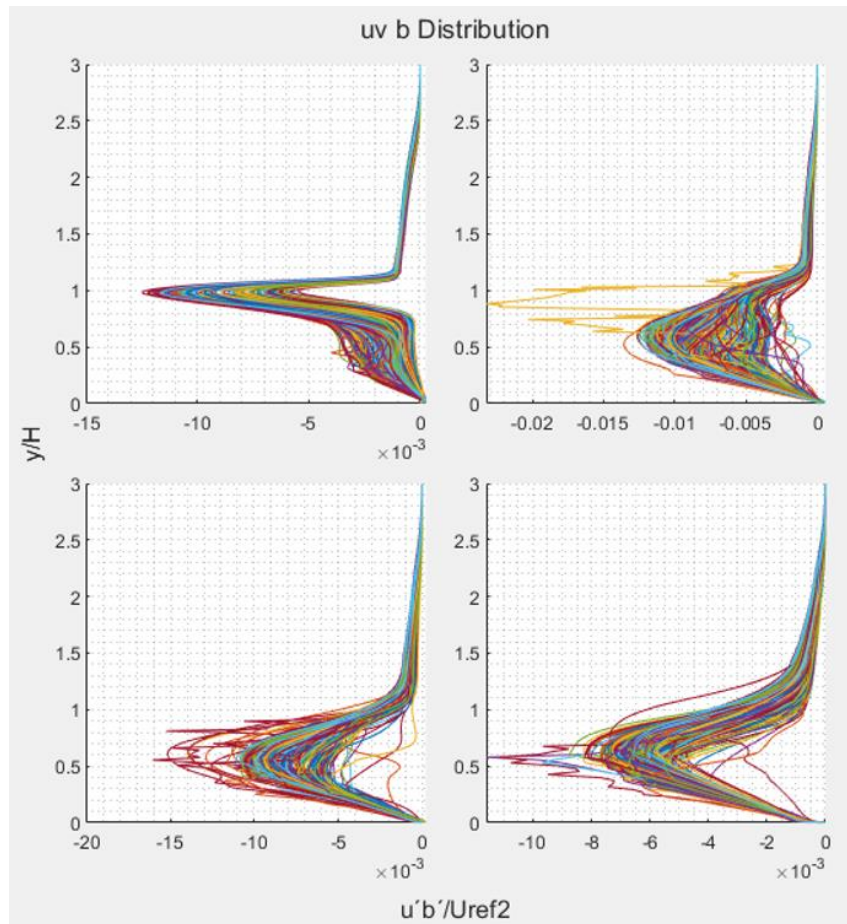


Figura 2-5: Distribución inicial de tensiones de Reynolds:

Método Propuesto – Inferencia Bayesiana

Este estudio puede ser claramente referenciado a otros anteriores, el modelo de cálculo de flujo turbulento SST es usado ampliamente en diversas aplicaciones y la necesidad de realizar una calibración al mismo, es una cuestión siempre presente. De tal manera, la **Inferencia Bayesiana** ha sido hallada como la **herramienta y el método adecuado** para realizar a la hora de abordar una calibración de este tipo.

En este punto, es necesario determinar en qué consiste esta calibración, como se lleva a cabo, que información es necesaria y que cuestiones se tendrán que tomar en cuenta.

En primer lugar, se puede decir que una **Calibración Bayesiana** es un **proceso iterativo**, cuyo propósito es disminuir las incertidumbres de los parámetros del modelo que se está calculando. El método no trabaja entre los valores calculados por el modelo y los

determinados experimentalmente, si no que determina la probabilidad de poseen los parámetros de entrada de otorgar un resultado de modelo con la menor incertidumbre posible respecto a los resultados experimentales [1].

Los **datos de entrada** necesarios para el método corresponden a la **muestra de datos a calibrar**, en el caso en estudio, esto serían las 300 combinaciones de coeficientes de corrección detallada anteriormente; un **modelo matemático representante** del fenómeno y una **muestra experimental fiable**, la cual en este caso es brindada por la NASA como muestra de validación.

En cuanto a la **metodología de calibración**, la misma se basa en tres puntos principales. Inicialmente, es necesario determinar una **función de probabilidad** que se ajuste a la **muestra de entrada**. En segundo lugar, **recopilar** una **base de datos de confianza** para poder utilizar de comparación. Por último, **calibrar** las funciones de probabilidad de entrada de los parámetros, **iterando** a través del Teorema de Bayes [2].

El segundo punto, suele ser desafiante para los estudios, ya que se corresponde a la parte experimental. En este caso, no lo es, ya que se corresponde al set de validación otorgado por la NASA.

Para **ajustar** la muestra inicial a una **función de probabilidad** conocida, si la misma se corresponde a **datos aleatorios** y cada parámetro puede ser **acotado** en un rango de valores, la misma puede ser considerada una **distribución uniforme** [3].

Respecto al tercer punto, es entonces donde se concentra el mayor desafío de esta calibración y donde los siguientes esfuerzos serán concentrados para poder determinar la mejor solución posible.

La calibración en sí, se puede enmarcar dentro de lo que se entiende como un **problema de inversión**, esto ocurre cuando **parámetros desconocidos no se pueden medirse directamente, sino que se estiman mediante datos experimentales** que solo se relacionen a los valores de entrada a través de un modelo matemático [4].

En tanto, para definir matemáticamente el método, es necesario comenzar por las bases del mismo. Esta corresponde al **Teorema de Bayes**, él se puede describir como a continuación [16]:

Sea $\{A_1, A_2, \dots, A_i, \dots, A_n\}$ un conjunto de sucesos mutuamente excluyentes y exhaustivos, tal que la probabilidad de cada uno es distinta a cero: $P(A_i) \neq 0$ para $i =$

$1, 2, \dots, n$.

Sea B otro suceso independiente tal que la probabilidad condicionada $P(B | A_i)$ es conocida, entonces la probabilidad condicionada $P(A_i | B)$ está definida por:

$$P(A_i | B) = \frac{P(B | A_i)P(A_i)}{P(B)} \quad (2.1)$$

Siendo la probabilidad condicionada $P(A | B) = \frac{P(A \cap B)}{P(B)}$

Lo que destaca de este teorema y su aplicación, es la posibilidad de **obtener la probabilidad de un suceso condicionado a otro a través de la probabilidad condicionada inversa y la probabilidad individual de cada suceso**. A priori, esto parecería un aumento en la dificultad, pero en ciertas situaciones, permite obtener información desconocida a través de otra más sencilla de obtener.

Una aplicación sencilla del mismo, pero que ayudara al lector a comprender el principio puede ser ejemplificado como a continuación [5]: Se realiza una encuesta a la población en general, sobre si estos practican algún deporte, los resultados de la misma se muestra en la figura 2-6 y a través de estos datos, se desea conocer, cual es la probabilidad de que un deportista sea hombre.

En otras palabras, se desea conocer $P(h | d) = \frac{P(h \cap d)}{P(d)}$

Directamente, observando los resultados de la encuesta, no es posible conocer dicho valor.

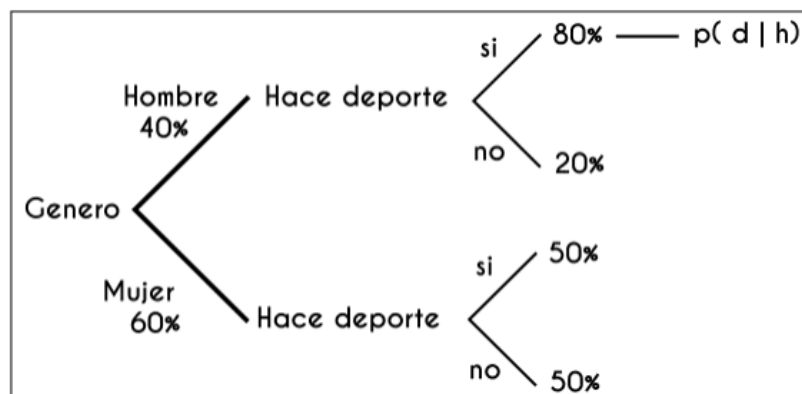


Figura 2-6: Valores de referencia problema de ejemplo [5]:

Allí, es donde la aplicación de la fórmula de Bayes toma importancia. Ya que mediante

esta se puede expresar.

$$P(h | d) = \frac{P(d | h)P(h)}{P(d)} = \frac{(0,8 * 0,4)}{(0,4 * 0,8) + (0,6 * 0,5)} = 0.516 = 51.6\%$$

Como se ha visto, se puede obtener una medición indirecta a través de mediciones directas. Esto da una idea de la utilidad y de porque es este teorema es la base principal del problema en estudio, ya que es prácticamente lo que se desea hacer: Obtener el camino inverso.

El concepto de **problema de inversión** se define cuando **parámetros desconocidos que no pueden medidos directamente, son estimados mediante datos experimental que solo se relacionan indirectamente con los parámetros a través de un modelo computacional**. Se le da el nombre de problema inverso, porque en lugar de propagar la información sobre los parámetros de entrada a través de un modelo computacional (el llamado enfoque hacia adelante), el objetivo es **propagar la información sobre las observaciones hacia atrás** para obtener información sobre las entradas del modelo [4].

Ampliando el desarrollo a un caso genérico:

Sea $X = \{x_1, \dots, x_n\}$ un vector el cual contiene las diferentes realizaciones independientes una de la otra, la cual se puede asociar una función de densidad de probabilidad $\pi(\cdot)$ y sea $\theta = \{\theta_1, \dots, \theta_n\}$ el vector aleatorio asociado que contiene los diferentes parámetros. Entonces, el **objetivo de la inferencia** es **determinar los parámetros θ que dan forma a la función de probabilidad**, expresado matemáticamente como: $\theta | X \sim \pi(\theta | x)$, representando a través de la linea vertical, la dependencia del valor de la izquierda, sobre el valor de la derecha.

Al tratarse de vector aleatorio, y usando diferentes técnicas, generalmente es posible enmarcar al vector de parámetros aleatorios Θ dentro de una función de probabilidad estadística, matemáticamente $\theta \sim \pi(\theta)$ y definida como la **Distribución Previa**.

Aplicando el Teorema de Bayes en base de funciones de probabilidad se obtiene:

$$\pi(\theta | x) = \frac{\pi(x | \theta)\pi(\theta)}{\pi(x)} \quad (2.2)$$

Para continuar con el desarrollo, es necesario introducir un nuevo concepto, el cual es el de **Función de Verosimilitud**.

Sea $X \rightarrow f(x, \theta), \theta = \{\theta_1, \dots, \theta_n\}$ una población, y $f(x_1, x_2, \dots, x_n, \theta)$ la función de

probabilidad o de densidad de una muestra aleatoria simple $X_1, X_2, \dots, X_n$ de X . Dada una realización, $x_1, x_2, \dots, x_n$, de la muestra, llamamos función de verosimilitud del parámetro a la función [6]:

$$L(x_1, x_2, \dots, x_n, \theta) = f(x_1, x_2, \dots, x_n, \theta) = f(x_1, \theta)f(x_2, \theta) \dots f(x_n, \theta)$$

La verosimilitud consiste en otorgar a un estimador/estimación una determinada "credibilidad", esto sería una mayor apariencia de ser el cierto valor (estimación) o el cierto camino para conseguirlo (estimador). En términos probabilísticos podríamos hablar de que **la verosimilitud es la probabilidad de que ocurra o se dé una determinada muestra si es cierta la estimación que hemos efectuado o el estimador que hemos planteado** [7]. En otras palabras, la verosimilitud mide cuán bien el modelo describe los datos que se han observado. En el contexto de la inferencia estadística, la verosimilitud se utiliza como una función de los parámetros del modelo y se maximiza para encontrar los valores óptimos de los parámetros.

Por lo tanto, si consideramos alas diferentes mediciones $X = \{x_1, \dots, x_n\}$ como realizaciones independientes de $\theta \mid X \sim \pi(\theta \mid x)$, es posible expresar la siguiente función de verosimilitud.

$$L: \theta \rightarrow L(\theta, X) \stackrel{\text{def}}{=} \prod_{k=1}^N \pi(x_k, \theta) \quad (2.3)$$

Es demostrable que es posible interpretar a la función de densidad de probabilidad, a través del Teorema de Bayes utilizando la siguiente relación. Dicha función será considerada como la **Distribución Posterior** y es la cual se obtendrá continuamente iterando.

$$\pi(\theta \mid x) = \frac{L(\theta; X)\pi(\theta)}{Z} \quad (2.4)$$

Donde Z es conocido como el termino *evidencia* y debe asegurar que la siguiente integral sea cercana a 1.

$$Z = \int_{\theta} L(\theta; X)\pi(\theta)d\theta \quad (2.5)$$

Ya que esta representa la probabilidad de que los datos observados sean generados por el modelo propuesto, y cuanto mayor sea la evidencia, más probable es que el modelo sea una buena representación de los datos. La evidencia es un factor de alta importancia

en la inferencia bayesiana, ya que permite comparar la probabilidad relativa de diferentes modelos frente a los datos observados.

Algo importante a remarcar, es que el computo de la distribución posterior no es trivial, sino que será necesario utilizar métodos de muestreos. Un método de muestreo, es una técnica utilizada para seleccionar una muestra representativa de una población o conjunto de datos más grande. En lugar de recopilar datos de toda la población, que a menudo es impracticable o costoso, se selecciona una muestra más pequeña para inferir información sobre la población en su conjunto.

En tal contexto, el **Método de Montecarlo basado en cadenas de Markov (MCMC)** es seleccionado, ya que experiencias previas muestran un excelente balance entre costo computacional y resultados obtenidos.

Para definir el mismo, se debe considerar un modelo matemático **M** que permite obtener cantidades de interés **y** como función de los parámetros de entrada **x**:

$$M: x \in D_x \subset R_N \rightarrow y = M(x) \subset R_N$$

Por lo tanto, considere los valores de entrada al modelo: $X = \{x_1; x_2, \dots; x_M\}$ y los resultados de salida del mismo $Y = \{y_1; y_2, \dots; y_N\}$, donde los parámetros de entrada no son medibles directamente. La **Calibración de Modelo Bayesiana** tiene como objetivo recuperar los datos de entrada X que den como respuesta los valores de salida Y . Este enfoque es conocido como el paradigma de **verificación y validación bajo incertidumbres**, y es básicamente la necesidad del trabajo realizado.

Si recordamos el Teorema de Bayes anteriormente mencionado y lo ponemos en funciones de las nuevas variables de interés $(\theta; X)$, es decir:

$$P(\theta | X) = \frac{P(X | \theta)P(\theta)}{P(X)} \quad (2.6)$$

Para el cálculo de la distribución posterior $P(\theta | X)$, haciendo uso del método MCMC, se generan muestras aleatorias de los parámetros θ , de acuerdo a una distribución de probabilidad que depende de las muestras anteriores, siendo una distribución uniforme la distribución inicial en el caso de estudio. En cada paso de la cadena de MCMC, se calcula la verosimilitud y se actualizan las creencias sobre los parámetros de acuerdo con el teorema de Bayes.

El algoritmo de MCMC comienza en algún valor inicial θ_0 y luego genera una nueva

muestra $\theta_{(i+1)}$ a partir de θ_i de acuerdo con una **distribución de transición** $Q(\theta_i, \theta_i + 1)$.

La **distribución de transición se calcula a través del modelo matemático** especificando una nueva distribución que determina la probabilidad de pasar de θ_i a $\theta_{(i+1)}$ y debe cumplir cierta condición para garantizar la convergencia de la cadena de Markov a la distribución posterior.

En cada iteración, se calcula la probabilidad de aceptación α de la nueva muestra $\theta(i + 1)$ según la siguiente fórmula:

$$\alpha = \min\left(1, \frac{P(\theta_i + 1 | X) * Q(\theta_i + 1, \theta_i)}{[P(\theta_i | X) * Q(\theta_i, \theta_i + 1)]}\right) \quad (2.6)$$

donde: $P(\theta | X)$ es la distribución posterior, $Q(\theta_i, \theta_{(i+1)})$ es la distribución de transición, y $P(X)$ se cancela en el numerador y denominador.

Si α es distinto a uno, es decir $\frac{P(\theta_i+1|X)*Q(\theta_i+1,\theta_i)}{[P(\theta_i|X)*Q(\theta_i,\theta_i+1)]}$ es menor a 1, se acepta la nueva muestra y se agrega a la cadena de Markov. De lo contrario, se rechaza la muestra y se mantiene la muestra actual θ_i en la cadena.

El proceso se repite hasta que se obtiene un número suficiente de muestras en la cadena de Markov y se verifica la convergencia de la cadena mediante diagnósticos estadísticos, como la autocorrelación y la traza.

Después de un número suficientemente grande de pasos, la cadena de MCMC converge a la distribución posterior deseada, y se pueden estimar estadísticas de interés, como la media o la mediana, de la distribución posterior, los cuales deben ser interpretados para obtener los parámetros que mejor ajusten la simulación a valores reales.

Modelo subrogado – PCE

En el desarrollo anterior, el lector podrá rápidamente identificar y comprender un **problema intrínseco del método** propuesto. Es necesario **iterar con un modelo matemático**. Sin embargo, resulta obvio que sería ilógico iterar con un modelo de CFD, el cual ya para resolverse una sola vez, ya posee un costo computacional altísimo y requiere horas de trabajo, lo que haría inviable tener que resolver el mismo ciento / miles de veces para realizar la inversión, llevando al desarrolló a ser teóricamente correcto pero imposible de aplicar.

Por lo tanto, es aquí donde el concepto de modelo subrogado o meta-modelo aparece. Este intenta compensar el aumento de los costes de la modelización estocástica sustituyendo los modelos computacionales demandantes por sustitutos simples de evaluar [8].

Existen diversos modelos subrogados desarrollados, y la elección de uno u otro, no se basa más que en la experiencia, practica y nivel de precisión requerido. Para este desarrollo, se decidió de utilizar un **Polinomio de Expansión del Caos (PCE)**, este meta-modelo fue utilizado en el pasado en aplicaciones similares en el Instituto y demuestra una precisión aceptable frente a un coste computacional relativamente bajo. Otros modelos fueron intentados de usar como el **Método de Kriging** o **Polinomio del Caos de Kriging (PCE-Kriging)**, los cuales prometen una mayor precisión, pero el mayor costo computacional que estos requieren hizo que su aplicación resulte inviable – *2% de avance en 24 horas de computo*-.

El **PCE** se construye a partir de un **conjunto de simulaciones de diseño previamente seleccionadas, en las que se evalúa el modelo original para diferentes combinaciones de valores de los parámetros de corrección de modelo**. A partir de estos datos, se ajusta un polinomio que aproxima la relación entre los parámetros de entrada y las salidas de interés, lo que permite la evaluación rápida del modelo subrogado.

Considere un vector $X \in R$ cuyos componentes son independientes uno del otro descriptos por la función de probabilidad de densidad (PDF) f_X y el modelo computacional $Y = M(X) \in R$, definiendo a la expansión polinomial del caos $M(X)$ como:

$$Y = M(X) = \sum_{\alpha \in N} y_{\alpha} \psi_{\alpha}(X) \quad (2.7)$$

Donde $\psi_{\alpha}(X)$ son polinomios multivariantes ortonormales respecto a f_X , $\alpha \in N$ el índice que los identifica, y mientras que $y_{\alpha} \in R$ los coeficientes del polinomio.

En aplicaciones reales, es necesario truncar el polinomio a sumas finitas, definiendo a $A \subset N$ como un set de índices de polinomios.

$$M(X) = M^{PC} \approx \sum_{\alpha \in A} y_{\alpha} \psi_{\alpha}(X) \quad (2.8)$$

Para la obtención de las bases de polinomios $\psi_{\alpha}(X)$ existen diversas posibilidades que

permiten la obtención de las mismas. A modo resumen, se podría mencionar las siguientes opciones en caso de que el lector desee ampliar: Polinomios de Hermite, Polinomios de Legendre y Polinomios de Chebyshev. En este caso, el programa computacional utilizado se encargará de la obtención de las mismas

Mas allá de la complejidad matemática de esta sección, se desea resaltar la necesidad tener que utilizar un **modelo subrogado**, ya que este **mejora significativamente la eficiencia computacional** y especialmente importante en aplicaciones como la en estudio donde el modelo original es costoso. Sin utilizar un modelo subrogado, este estudio no se podría llevar a cabo con recursos computacionales normales y los esfuerzos requeridos llevarían a que el modelo de flujo turbulento SST no se utilizado, ya que calibrarlo sería una tarea demasiado costosa.

En definitiva, el **modelo subrogado** no representa ni más ni menos que una **función**, para la misma la entrada serán 5 valores numéricos correspondientes a cada uno de los parámetros, mientras que la salida del modelo, será cada uno de los resultados de interés, ya sea así velocidad, coeficientes de presión o rozamiento o tensiones de Reynolds.

Calibración

En este capítulo se pretende describir todo el proceso de calibración realizado, esto incluirá diferentes desde la selección del programa computacional utilizado, el tratamiento de datos y la lectura de resultados.

Lo que se desarrollara como un proceso sencillo y sin complicaciones, no fue así, si no todo lo contrario. Esta etapa fue altamente desafiante atravesando diferentes altibajos para poder arribar a los resultados que serán detallados.

Cuestiones como una temática a la cual era totalmente ajeno al comienzo del estudio y la utilización de un software potente, pero a la vez relativamente nuevo, con escaso soporte y una comunidad no muy extensa de usuarios, hicieron que cada obstáculo presentado sea un completo desafío a resolver.

Herramienta computacional – UQLab

Desarrollos como el llevado a cabo en el presente informe y similares, se encuentran todo el tiempo en la ingeniería y en la comunidad científica. Cuestiones a trabajar con métodos de muestreo, calibraciones bayesianas, metamodelos, entre otros, existen desde hace años.

En tal sentido, en el año 2013, Prof. Bruno Sudret de la Cátedra de Riesgo, Seguridad y Cuantificación de la Incertidumbre del ETH de Zúrich, se propone desarrollar un software que se encargue de tratar estas problemáticas. Siendo lanzado en el año 2015, **UQLab** se presenta como un software de **código abierto basado en Matlab**, el cual propone una serie de funciones complementarias con el propósito de atender temas relacionados a la cuantificación de incertidumbres [9].

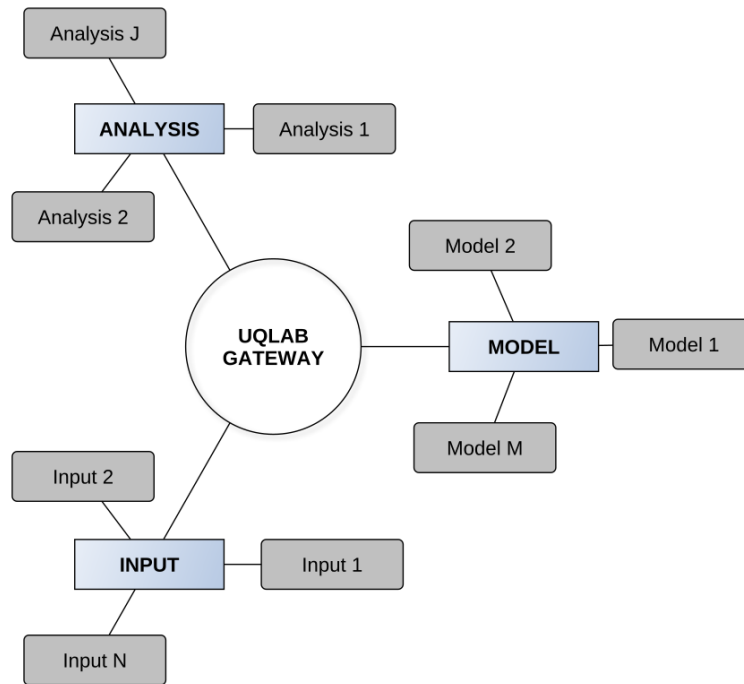


Figura 3-1: Diagrama UQLab, entradas y salidas [9]:

La elección de llevar a cabo la presente calibración mediante una inferencia bayesiana, se debe no solo a la capacidad de método para realizar la misma, sino también a la existencia de UQLab.

La inferencia bayesiana es un método con altas probabilidades de arribar a valores correctos para la calibración buscada. Sin embargo, de poco nos serviría si no contáramos con un software que nos permita llevar a cabo la misma. Vale aclarar que programar una calibración mediante inferencia bayesiana como esta desde cero, constituye una tarea de altísima dificultad, demandando altos conocimientos en programación, esfuerzos personales y tiempo. Ya que no solo sería programar la inversión, si no también, método de muestreo, modelo subrogado y todos los tratamientos de datos intermedios. Tampoco valdría la pena y si esta fuera la única opción, utilizar un modelo como el SST que precisa de una calibración de este tipo, carecería de total sentido, ya que los resultados obtenidos no compensarían el esfuerzo demandado.

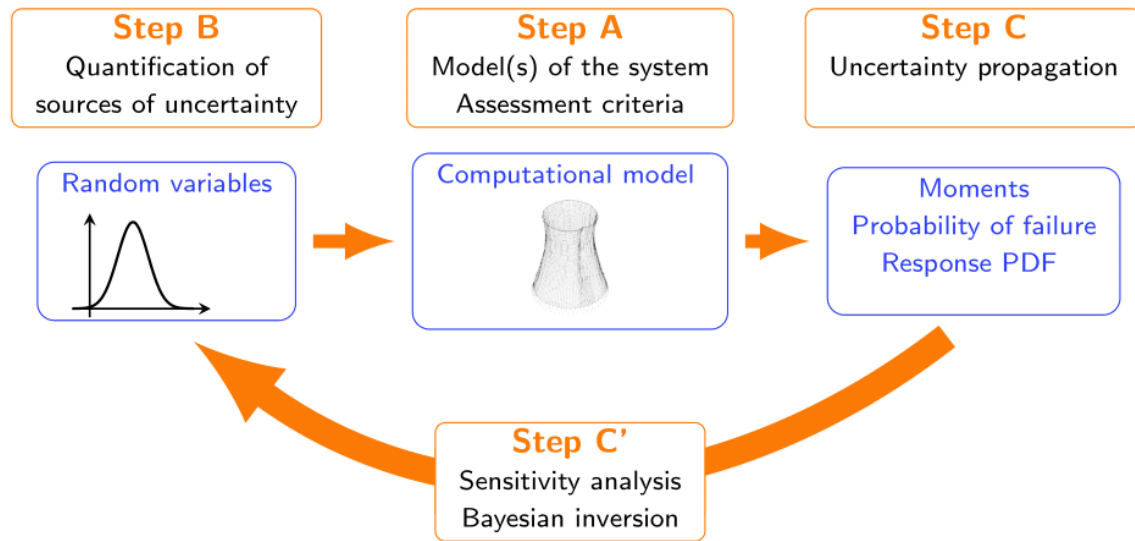


Figura 3-2: Flujo de datos en UQLab [9]:

Por lo tanto, se podría decir, que la elección de **UQLab como herramienta** computacional y la selección de realizar una calibración mediante **inferencia bayesiana** fue una **decisión conjunta**. Se optó por realizar una inferencia bayesiana porque existe UQLab para procesarla, sino existiera UQLab, se hubiera optado por otro método, o bien la calibración no sería posible.

Desarrollo del código

Para la escritura del código, principalmente se pueden identificar tres etapas las cuales serán detalladas a continuación.

En un primer lugar, es el ingreso de la enorme base de datos inicial presentada y el proceso para estructurar los datos que son de interés y útiles para el estudio en el formato que UQLab lo requiere. Luego, la etapa de calibración en sí misma y, por último, la lectura y procesamiento de los resultados obtenidos.

Carga de datos

Esto es algo que a priori se supone sencillo, pero dada la cantidad y diversidad de la información otorgada inicialmente, el trabajo requiere de cierto nivel de análisis para resultar exitoso.

La base de datos otorgada se conforma principalmente de 302 archivos de datos de Excel. El primer archivo cuenta con un arreglo de 300 x 5, donde cada fila representa una combinación de 5 parámetros. Cada fila del primer archivo, se corresponde a uno de los otros 300 archivos.

	A	B	C	D	E	F	G	H
1	PG	SARC_C1	SARC_C2	SARC_C3	SAS	SHA1		
2	5	0.5	2	0.2142857143	0.6363636364	64fd08922dee4e25dd751ac695877ed282edabfd		
3	2.5	1	4	0.4285714286	1.272727273	0362336b9b521577a95063f518d1c37536f86cc5		
4	7.5	0.1666666667	6	0.6428571429	1.909090909	774f0ce8b225b8f8d7cdf05d63c033f223f3c94		
5	1.25	0.6666666667	8	0.8571428571	2.545454545	8568d776bcccc00e52658c4bb8f16459d624f393		
6	6.25	1.166666667	0.4	1.071428571	3.181818182	15d977a050f2b7750122842227508c33c58ec73a		
7	3.75	0.3333333333	2.4	1.285714286	3.818181818	994d302e7be8141f662c972a4c89141df735824d		
8	8.75	0.8333333333	4.4	0.0306122449	4.454545455	2c16c59dc5811b4dd84682d053b6442014e68dbf		
9	0.625	1.333333333	6.4	0.2448979592	5.090909091	6fc940928b63c88674b83fdb329550a6c663457		
10	5.625	0.0555555556	8.4	0.4591836735	5.727272727	f48a801af2010a68f211a94ab9bdedaabfd2790f		
11	3.125	0.5555555556	0.8	0.6734693878	6.363636364	f5743deac00818cb7b10575aaf2a2ae30cfa5072		
12	8.125	1.055555556	2.8	0.887755102	0.05785123967	61f94bf5feb5ce40cceb47f6656643603ad6a2f2		
13	1.875	0.2222222222	4.8	1.102040816	0.694214876	3509ad819e686a7bf7d3f977d3473e5700407a07		
14	6.875	0.7222222222	6.8	1.316326531	1.330578512	b361ae5f69728dd55c975147cd52650c5c6db732		
15	4.375	1.222222222	8.8	0.0612244898	1.966942149	b983cde72dc5879c7d094e8d945a38531c0c57de		
16	9.375	0.3888888889	1.2	0.2755102041	2.603305785	ba89664d518bbfb7f536ec27b6d3fc1074b71e40		
17	0.3125	0.8888888889	3.2	0.4897959184	3.239669421	ebef34764784dd68db8ce081aa87b9c13fa76670		
18	5.3125	1.388888889	5.2	0.7040816327	3.876033058	599ca1d4b150bd8c8559c549b54888beb0071e402		
19	2.8125	0.1111111111	7.2	0.9183673469	4.512396694	0e1f5cc779efceacdf1e9619fe2ad8be7bb9f57a		
20	7.8125	0.6111111111	9.2	1.132653061	5.148760331	f00101c5eee208841c2f39878734bcae2568ed26		
21	1.5625	1.111111111	1.6	1.346938776	5.785123967	deb08f077c0e3ff7e06c5d54082541bda990f33b		
22	6.5625	0.2777777778	3.6	0.09183673469	6.421487603	13de644039587d26dfb8407518ea387be250b32a		
23	4.0625	0.7777777778	5.6	0.306122449	0.1157024793	be8e3d41f9173d82b17707df95bacaf1c4adcc6b		
24	9.0625	1.277777778	7.6	0.5204081633	0.7520661157	53614facb2c84954872c5d7f4486e47d07f65f7		
25	0.9375	0.4444444444	9.6	0.7346938776	1.388429752	0e91cc41782a21ccab20ec8321cf904c9ebe0d1c		

Figura 3-3: Archivo de combinaciones de parámetros:

La figura 3-3, muestra las primeras 25 columnas donde se puede observar lo detallado anteriormente y a su vez, la aleatoriedad de los valores por los que se optó.

En cada uno de los siguientes 300 archivos, se puede encontrar un arreglo de 1162 x 6, donde las dos primeras columnas corresponden a valores de posición X e Y/Z y las otras cuatro a valores de velocidad, tensiones de Reynolds y coeficientes de presión y rozamiento, es decir, las variables de salida del modelo. En la siguiente imagen, se puede

observar un ejemplo de esto y su formato, siendo la planilla encabezada por el identificador de la configuración de parámetros que le corresponde y continuando por los resultados de simulación.

	A	B	C	D	E	F
1	34caea8415d5db003405f9fa2f53c0419db89634					
2	X	Z	x_velocity	cp	uw_b	cf
3	1	0	0	-0.248491	0	0
4	1	9	0	-0.11994	0	0
5	1	8.99969	0.0162752	-0.119939	3.94E-10	0
6	1	8.99934	0.0347401	-0.11994	1.84E-08	0
7	1	8.99894	0.0557077	-0.119939	2.54E-07	0
8	1	8.99848	0.0795065	-0.119939	1.86E-06	0
9	1	8.99797	0.106449	-0.119939	9.25E-06	0
10	1	8.99739	0.136677	-0.119939	3.47E-05	0
11	1	8.99672	0.169807	-0.119939	0.000100686	0
12	1	8.99597	0.204537	-0.119939	0.000227589	0
13	1	8.99512	0.238806	-0.119939	0.000407197	0
14	1	8.99416	0.270679	-0.119939	0.000611238	0
15	1	8.99306	0.299623	-0.119939	0.000769166	0
16	1	8.99182	0.325951	-0.119938	0.000894973	0
17	1	8.99041	0.349964	-0.119938	0.00099681	0
18	1	8.98881	0.371974	-0.119937	0.00107785	0
19	1	8.98699	0.392316	-0.119937	0.00114268	0

Figura 3-4: Archivo ejemplo de combinación de resultados de simulación:

El ultimo archivo, corresponde a una arreglo de 165 x 6, el cual contiene los resultados reales otorgados por la NASA, organizados de la misma manera que los archivos anteriores, el cual fue descargado directamente desde la misma web de la agencia, presentes en los siguientes enlaces: [Backstep Cp data](#); [Backstep Cf data](#); [Backstep profiles](#) y agrupado en una única planilla.

El primer paso a realizar es cargar toda esta información a Matlab para poder trabajar con la misma. De tal manera, del primer y último archivo, se obtienen directamente dos matrices que se corresponden directamente con los datos de las tablas de Excel y ya se encuentran listas para su uso. En cuanto a las 300 planillas de resultados, se ordenan en una estructura de 300 x 1 que contiene a los 300 arreglos y las cuales deberán ser post procesados para adaptarlo a una forma con la cual UQLab pueda trabajar.

Al conocer cómo trabaja la inferencia bayesiana, se puede comprender que la misma trabaja comparando valores de modelo con la base de valores reales otorgados. La base de datos de valores de modelo, es enormemente amplia, la misma cuenta con resultados para diversas posiciones, incluso para algunas o en su mayoría afuera del intervalo del que se cuentan datos reales.

Lamentablemente es necesario descartar todo valor de modelo que se encuentre por fuera de los intervalos entre los cuales se cuenta con información real. No es lo que uno desearía, ya que obtener una base de datos tan amplia y resultados para diversas posiciones demandó un esfuerzo computacional y tiempo mayor que si solo se hubiera realizado desde un principio para los intervalos en los que se cuenta información. En la siguiente tabla se muestran los intervalos y posiciones sobre las cuales se cuenta con información experimental. Intervalos y posiciones a los cuales deberá ser reducido el resto de los resultados.

Tabla 3-1: Intervalos iniciales de resultados.

Variable	Posición/Intervalo	
	X	Z
Vel	[1;4;6;10]	[0 a 8.2]
uw_b	[1;4;6;10]	[0 a 8.2]
Cp	[-8.5 a 37.5]	0
Cf	[-4 a 36]	0

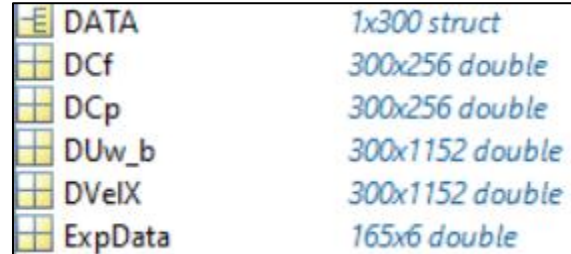
De todas formas, es sencillo de comprender que la obtención de parámetros reales (involucrando trabajo de laboratorio y experimentos) representa un desafío mucho mayor que obtener resultados de modelo. Además, teniendo en cuenta que dichos resultados son lo que respaldaran la calibración y al ser obtenidos por una agencia de renombre como la NASA, no tendría ningún sentido tratar de replicarlos o extender la lista.

Al llevarse a cabo dicho reordenamiento, de los resultados de modelo iniciales, la misma se modificó quedando como se observa en la tabla 3-2 a continuación:

Tabla 3-2: Intervalos útiles de trabajo de resultados.

Variable	Posición/Intervalo	
	X	Z
Vel	[1;4;6;10]	[0 a 3]
uw_b	[1;4;6;10]	[0 a 3]
Cp	[-5 a 30]	0
Cf	[-4 a 30]	0

Siendo esta la base de datos que se utilizara en el próximo apartado para realizar la calibración. Se muestra en la figura 3-5 como quedan los arreglos en Matlab, siendo este el formato que UQLab requiere, correspondiendo cada columna a todos los resultados de una simulación para una de las variables de salida según el arreglo.



DATA	1x300 struct
DCf	300x256 double
DCp	300x256 double
DUw_b	300x1152 double
DVelX	300x1152 double
ExpData	165x6 double

Figura 3-5: Arreglos finales en MatLab:

Pasos de calibración

A continuación, se detalla el proceso de calibración y como se trabajó con UQLab para llevar a cabo la mismo. Mientras en el capítulo anterior se desarrolló el proceso en forma teórica, en este apartado se busca mostrar cómo se trabajó con el software.

El **primer paso** a realizar corresponde a definir una **distribución previa de variables** a calibrar. En este caso, la misma se corresponde al **arreglo de 300 x 5** donde se encuentran las 300 combinaciones de **parámetros utilizados** en las simulaciones.

Vale aclarar que UQLab presenta la posibilidad de determinar un arreglo de variables aleatoriamente según un tipo de distribución deseada, puede ser gaussiana, uniforme, triangular, o cualquier otra que la estadística contemple.

La importancia de lo anterior, destaca la **capacidad que el software posee de calibrar directamente un modelo el cual no ha sido ejecutado anteriormente**. En el caso en estudio es algo diferente, ya que la calibración en sí, si es al modelo, pero se realiza indirectamente a través de los parámetros y, además, al utilizar un modelo subrogado, impide dicha alternativa.

Entonces, es así como la **matriz de parámetros** pasa a conformar lo que se conoce como **distribución previa o primaria**.

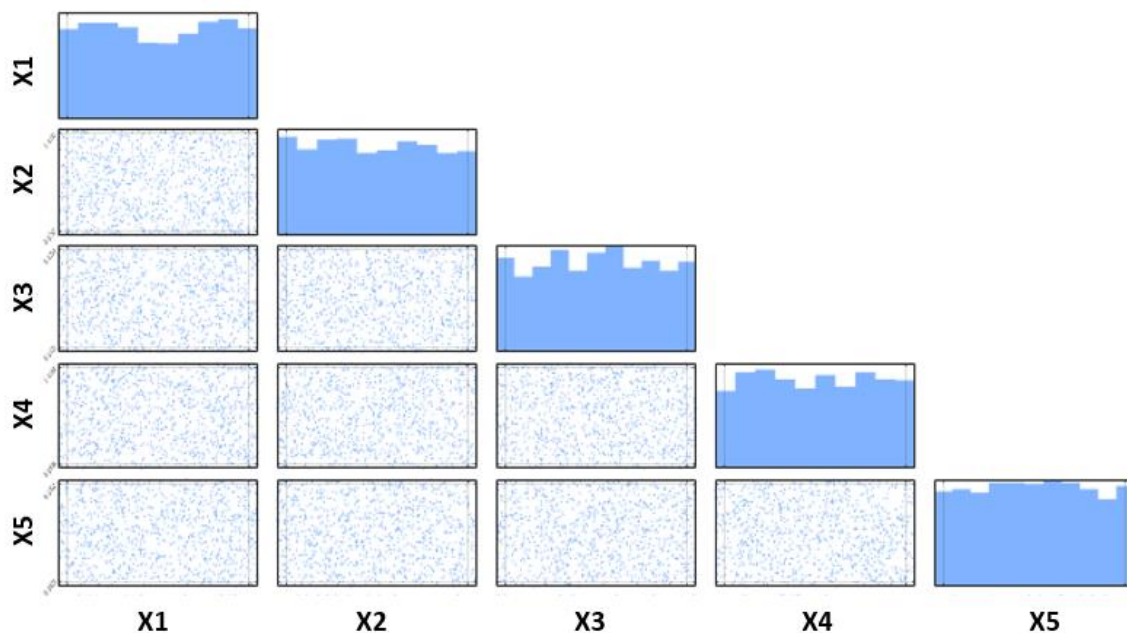


Figura 3-6: Distribución primaria:

En la figura 3-6 es posible observar la distribución uniforme de variables, esto se aprecia en la diagonal superior de imágenes. Las demás graficas corresponden a probabilidades condicionadas entre variables, las cuales no tienen mucha importancia en este momento.

El siguiente paso en el proceso corresponde a determinar el modelo matemático que será utilizado en las posteriores iteraciones. Como ya fue explicado en secciones anteriores, se utiliza un Polinomio del Caos Expansivo (PCE) como modelo subrogado.

El **PCE** es **calculado** a través de **valores de entrada y salida del modelo original**. Es decir, las variables de entrada a lo que será nuestro modelo subrogado, son los valores previamente simulados.

Por lo tanto, las funciones de UQLab que desarrollan el PCE, necesitan de la matriz de parámetros y de los resultados de simulaciones. Todos previamente estructurados en el formato que el programa lo requiere. Esto se carga al software con las funciones `MetaOpts.ExpDesign.X` e `MetaOpts.ExpDesign.Y`, para luego obtener el modelo utilizando `uq_createmodel(MetaOpts)`.

Como se aclaró previamente, un modelo subrogado no es más que una función. En este caso una función de 5 variables, un valor por cada parámetro de corrección que posee el modelo inicial. Al estar trabajando con **4 resultados diferentes** (coeficiente presión y rozamiento, velocidades y tensiones de Reynolds), se debió por crear **4 PCE distintos**, uno para cada resultado.

Lo siguiente fue especificar la data de experimental para la optimización de bayesiana, colocar todos los datos en el formato requerido y correr el programa.

En el anexo 1 adjunto al final de este informe se encuentra el código de Matlab / UQLab utilizado, donde se podrán contemplar los pasos detallados y observar el formato requerido por UQLab para realizar la inversión. En las siguientes figuras, se muestra la salida inicial del programa y se realizan algunos comentarios sobre los datos resaltados.

```

The discrepancy was not specified,
using unknown i.i.d. Gaussian discrepancy...
The solver was not specified, using MCMC
The sampler was not specified, using affine invariant ensemble samg
Starting AIES...

|#####| 100.00%

Finished AIES!

%----- Inversion output -----%
Number of calibrated model parameters: 5
Number of non-calibrated model parameters: 0

Number of calibrated discrepancy parameters: 275

%----- Data and Discrepancy
% Data-/Discrepancy group 1:
Number of independent observations: 1

Discrepancy:
Type: Gaussian
Discrepancy family: Row
Discrepancy parameters known: No

Associated outputs:
Model 1:
Output dimensions: 1
to
112

Associated outputs:
Model 2:
Output dimensions: 1
to
32

% Data-/Discrepancy group 3:
Number of independent observations: 1

Discrepancy:
Type: Gaussian
Discrepancy family: Row
Discrepancy parameters known: No

Associated outputs:
Model 3:
Output dimensions: 1
to
111

% Data-/Discrepancy group 4:
Number of independent observations: 1

Discrepancy:
Type: Gaussian
Discrepancy family: Row
Discrepancy parameters known: No

Associated outputs:
Model 4:
Output dimensions: 1
to
20

```

```

%----- Solver
Solution method: MCMC

Algorithm: AIES
Duration (HH:MM:SS): 00:53:16
Number of sample points: 3.00e+04

%----- Posterior Marginals
-----
| Parameter | Mean | Std | (0.025-0.97) Quant. | Type |
-----
| X1 | 6 | 0.29 | (5.4 - 6.6) | Model |
| X2 | 1.2 | 0.073 | ( 1 - 1.3) | Model |
| X3 | 8.6 | 0.46 | (7.5 - 9.4) | Model |
| X4 | 1.4 | 0.049 | (1.3 - 1.5) | Model |
| X5 | 0.17 | 0.073 | (0.015 - 0.3) | Model |

```

Figura 3-7: Salida primaria de UQLAB:

En primer lugar, es posible observar que UQLab otorga una revisión de los datos con los que se está trabajando, aclara el método de muestreo utilizado: En este caso MCMC. Sigue detallando la cantidad de salida de modelo y su dimensión.

Muestra el **tiempo de procesamiento**, en este caso fue de **53 minutos** y como resultado por defecto en una calibración muestra la media de parámetros.

Aclaración: Los términos X1, X2, X3, X4 y X5 corresponden cada uno a un valor de $(C\omega_4 - cr1 - cr2 - cr3 - \zeta_2)$ en el mismo orden que se encuentran ambas sucesiones.

Postproceso de resultados

De los resultados indicados en la figura 3-7, respecto a los parámetros a calibrar, se otorga la media de los mismos, lo cual puede representar un valor de interés o no. En el caso en estudio, interesa obtener los valores de parámetros con la mayor posibilidad de ser los que otorguen resultados correctos en la simulación. En otras palabras, seleccionaremos los valores con mayor probabilidad y no lo que represente la mediana. Esto se puede ver la figura 3-8, donde se encuentra el resultado de la inversión es decir la distribución posterior y de la cual se toman los valores con mayor probabilidad.

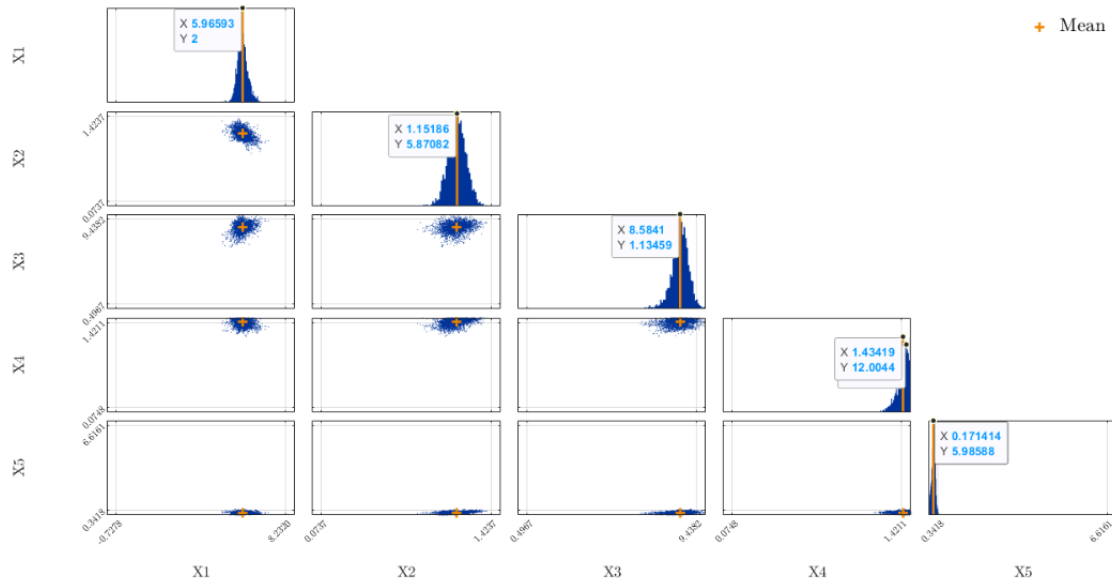


Figura 3-8: Distribución posterior de parámetros calibrados:

Por lo tanto, los valores obtenidos de la calibración son los siguientes:

- $C\omega 4 = 5.96$
- $cr1 = 1.15$
- $cr2 = 8.58$
- $cr3 = 1.44$
- $\zeta 2 = 0.17$

En este punto, ya se podría pasar a una etapa de validación, donde se realizaría una nueva simulación CFD del problema, utilizando el modelo SST con los nuevos valores de parámetros y así comprobar si la calibración fue efectiva o no.

No obstante, al poseer los 4 modelos subrogados, los cuales representan no más ni menos que una función que otorga resultados en base a una entrada de 5 parámetros, es interesante observar que respuesta otorga el mismo. Si bien, no es una solución precisa como al correr nuevamente una simulación, los PCE son creados en base a otras 300 simulaciones lo cual da un respaldo a los valores que la nueva evaluación otorgue.

Considerando que el costo computacional y el esfuerzo requerido es bajo, se evalúan en base a los parámetros calibrados, los 4 PCE y se muestran los resultados que se obtienen a continuación:

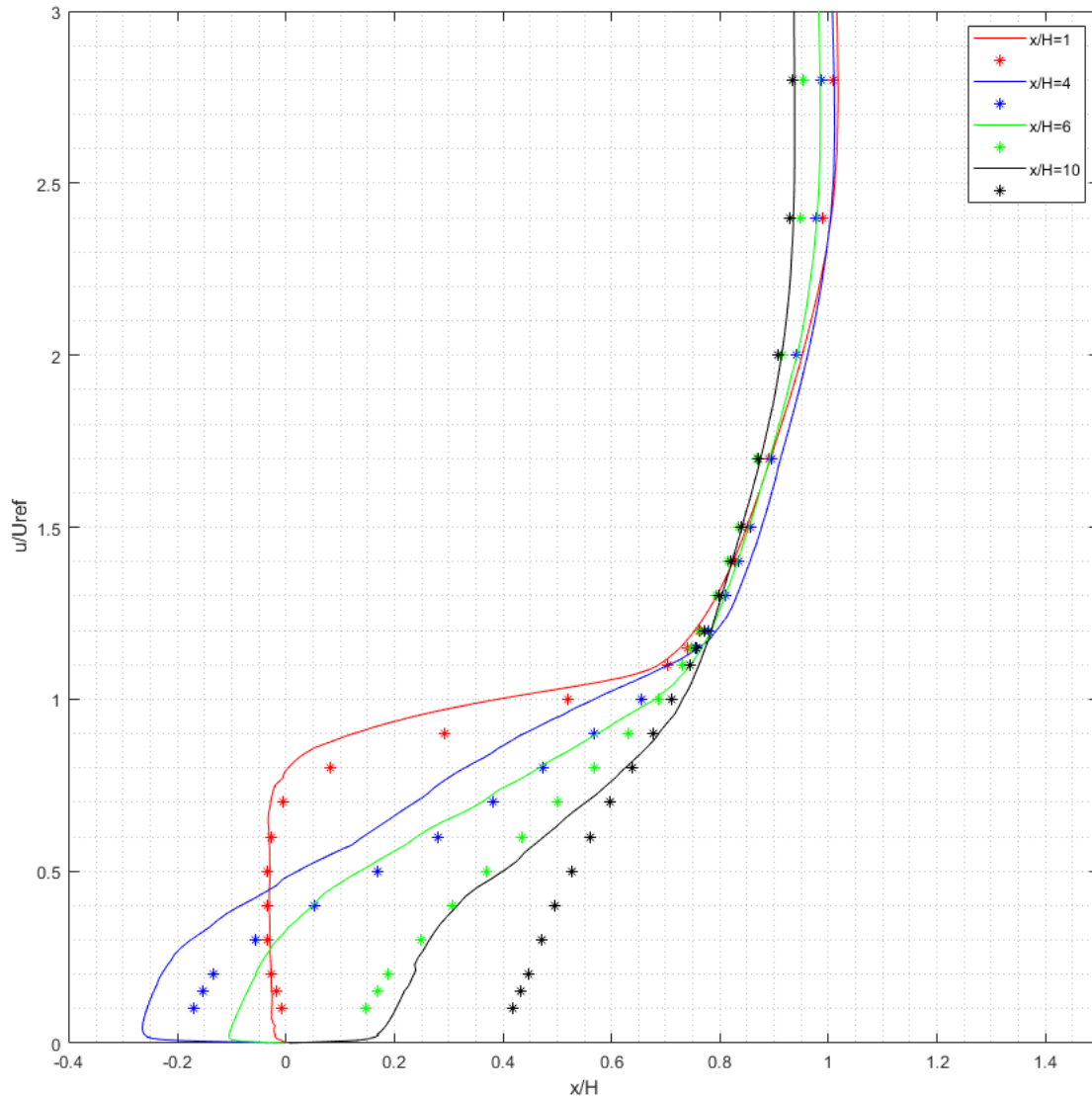


Figura 3-9: PCE de velocidad evaluado en valores calibrados:

La línea continua representa la respuesta del modelo subrogado, mientras que los asteriscos representan los valores experimentales otorgados por la NASA. Si comparamos los mismos directamente puede ser un poco decepcionante ya que los resultados no coinciden completamente.

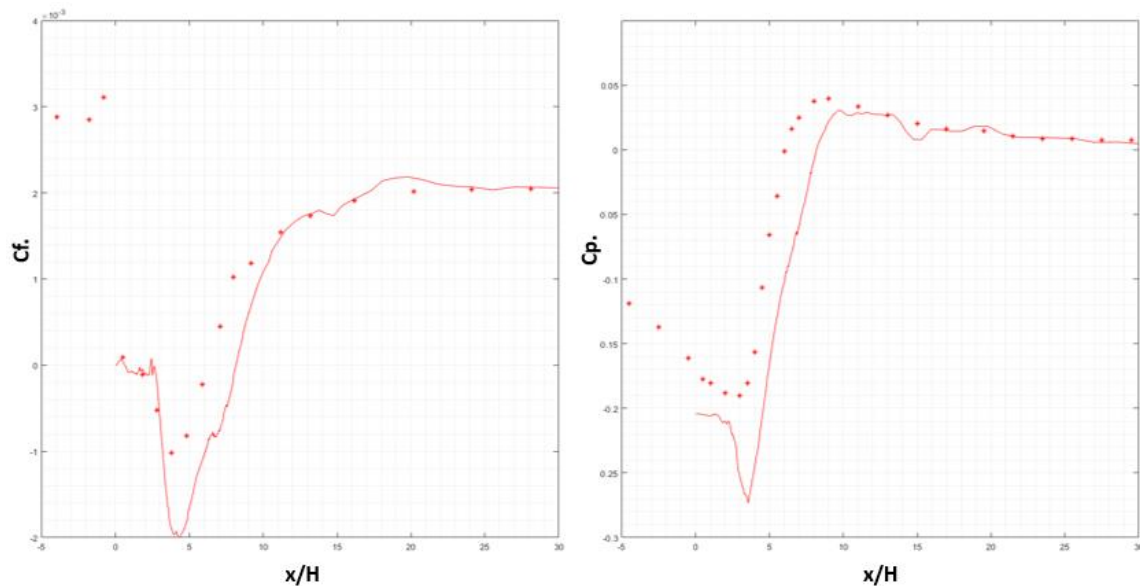


Figura 3-10: PCE de C_p y C_f evaluado en valores calibrados:

Sin embargo, si recordamos las figuras 1-2 es posible observar que, en los datos de simulación obtenidos por la NASA, estos no coinciden perfectamente con los datos experimentales y a su vez, se parecen a los obtenidos al evaluar los modelos subrogados, lo que nos otorga altas expectativas sobre los resultados obtenidos y la futura validación a realizar.

Es necesario reconocer que el modelo de SST del programa DLR TAU-CODE, no funcionara mejor que el simulador de la NASA, por lo que obtener resultados semejantes a los que la NASA obtuvo en su simulación, harían del presente desarrollo un éxito total y serían resultados con los que desde el Instituto se estaría más que conforme.

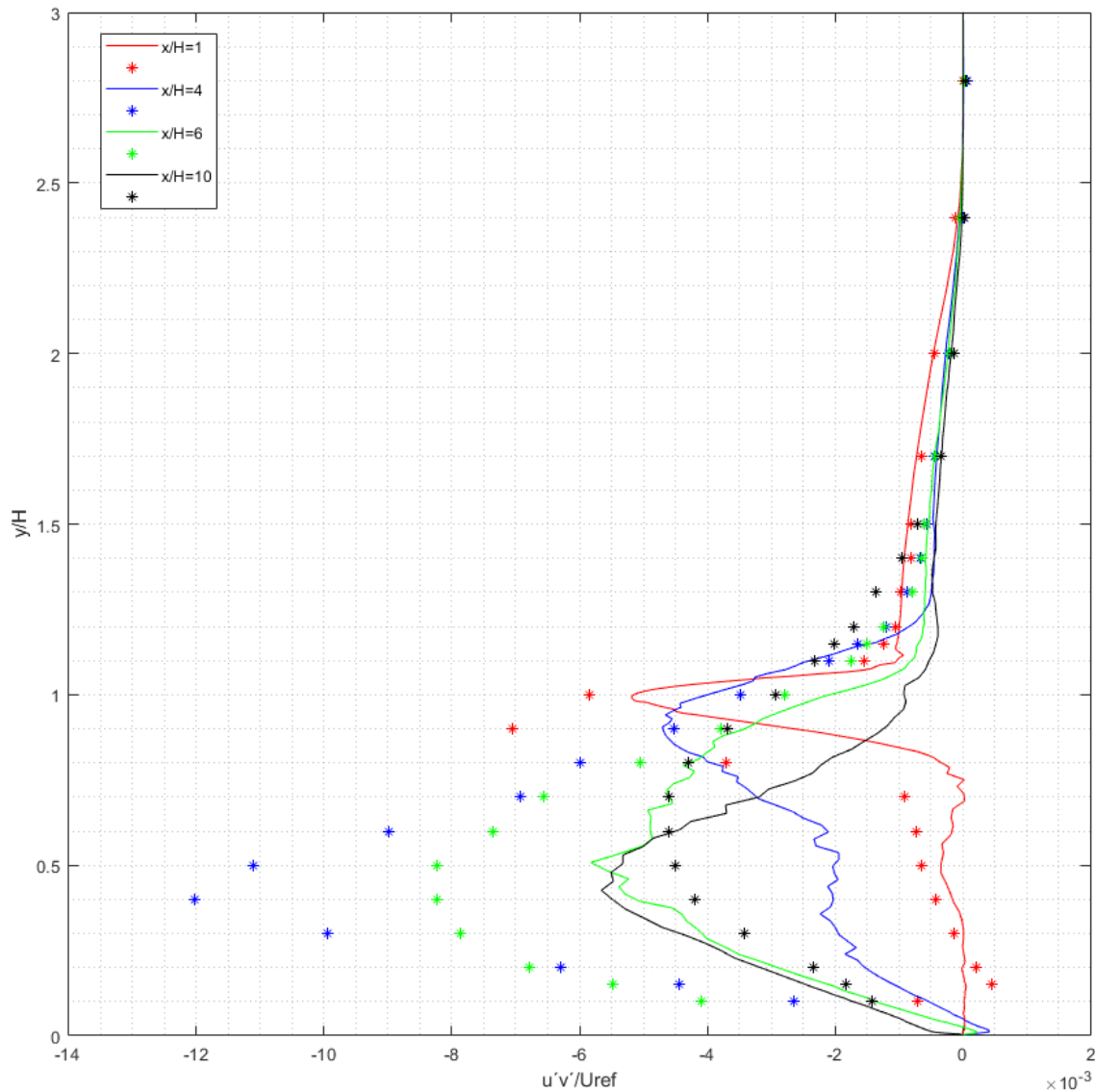


Figura 3-11: PCE de tensiones de Reynolds evaluado en valores calibrados:

Respecto a la figura 3-11, en la misma se observan los resultados obtenidos al evaluar el modelo subrogado para el cálculo de tensiones de Reynolds. En este caso los resultados no son tan positivos, ya que no se asemejan ni a los experimentales ni a los simulados. Además, se pueden observar oscilaciones en los resultados, las cuales son lógicamente no deseable y otorgan un indicio de irregularidades en los resultados. De todas formas, esto es algo que ya se conocía de antemano y el presente desarrollo, especialmente la etapa de validación ayudara a dimensionar el error obtenido.

Validación

En esta sección, se procederá a realizar nuevamente la simulación utilizando los parámetros calibrados. Se comenzará con una breve introducción al software de CFD empleado. A continuación, se presentarán las simulaciones, seguido por un análisis de los modelos, se destacarán los datos relevantes en los códigos utilizados, se discutirán las variables de interés y, por último, se llevará a cabo un estudio de los resultados obtenidos.

Vale aclarar que los modelos ya se encontraban desarrollados previamente al momento de la obtención de los datos iniciales. Por lo que en esta etapa no se debe esperar un desarrollo completo de los mismos. En cambio, la intención es poder interpretar los diversos códigos utilizados y estudios realizados, comprender las razones que llevaron a la elección de un modelo u otro y, sobre todo, interpretar los resultados obtenidos.

DLR TAU-Code

El software utilizado para la resolución del problema y el cual se desea verificar su funcionamiento es el DLR TAU-Code. El mismo es un desarrollo en conjunto del DLR (Centro Aeroespacial Aleman) y la Universidad Tecnica de Braunschweig, este comenzó en la década de los 80 y es un desarrolló en curso hasta el día de hoy. Con el transcurso del tiempo otras instituciones fueron realizando aportes y tomando parte en el desarrollo del código [12].

Un programa de software de CFD se divide principalmente en tres componentes principales. En primer lugar, está la etapa de preprocesamiento, donde se realiza la malla o discretización de un modelo CAD para obtener un dominio discreto. Además, se definen las condiciones de contorno y flujo que determinarán la situación del estudio del flujo. Estos datos conforman la entrada del modelo.

El segundo componente es el simulador de flujo en sí mismo. Este programa se encarga de iterar a través de cálculos numéricos utilizando un método específico para obtener los resultados deseados.

Por último, se requiere un módulo de adaptación que ajusta e interpola los datos para refinar y crear una solución coherente. Este módulo es esencial para obtener resultados más precisos y confiables en el estudio de CFD [10].

El código utilizado en el estudio de CFD muestra una amplia versatilidad en términos de formatos de entrada de modelos CAD y mallas. Esto permite que se puedan utilizar diversos programas de modelos CAD para realizar las simulaciones [11]. Esta capacidad es de gran interés, ya que en el estudio actual se ha recibido un modelo mallado proporcionado por la NASA, lo cual nos brinda la confianza de que está correctamente realizado y será consistente con los resultados encontrados en su sitio web.

Modelo CAD y mallado

Como ha sido mencionado, el primer paso para llevar a cabo una simulación de CFD es obtener un modelo CAD de la pieza o entorno que se desea estudiar, y luego generar una malla que discretice el dominio.

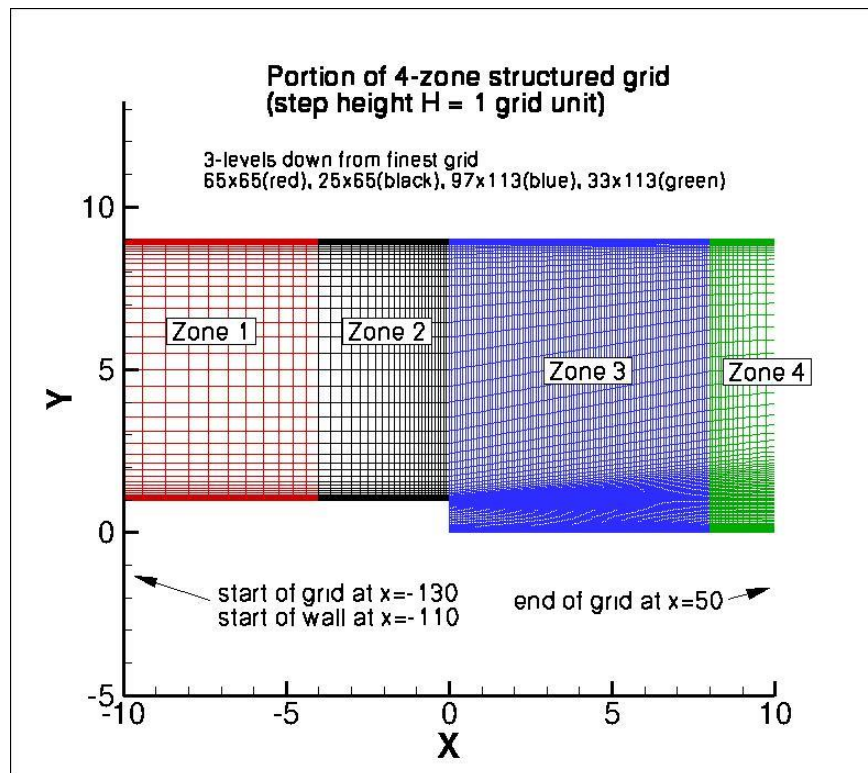


Figura 4-1: Modelo mallado otorgado por la NASA [13]:

En este caso particular, la NASA proporciona un modelo CAD ya mallado con distintos niveles de refinamiento que se pueden utilizar. La malla se compone principalmente de cuatro zonas diferentes, cada una con un grado de refinamiento mayor o menor, dependiendo de la precisión necesaria y complejidad de los fenómenos involucrados en cada área. Esto permite que los recursos computacionales se enfoquen en las zonas que son más relevantes para el estudio

El mallado adecuado desempeña un papel fundamental en las simulaciones CFD, ya que sin este no se podría lograr precisión y confiabilidad de los resultados. Un mal mallado puede introducir errores numéricos y afectar la representación precisa del flujo y los fenómenos asociados. Un mallado de alta calidad garantiza una representación adecuada de la geometría y las características del flujo, permitiendo una resolución adecuada de los gradientes y patrones de flujo. Además, un mallado bien estructurado y refinado correctamente puede capturar de manera efectiva los detalles importantes del flujo [11].

No se pueden esperar buenos resultados con un mallado deficiente. Es por esto que tener la facilidad de contar con un buen mallado directamente otorgado por la NASA presenta una gran ventaja, ya que garantiza la calidad y la confiabilidad. Esto es otra condición que colabora a que el caso elegido para realizar la comprobación del modelo SST sea el correcto.

Pasos de simulación

Una vez que se ha generado el modelo mallado, el siguiente paso implica seleccionar un solucionador, desarrollar el código correspondiente, ejecutar el programa y analizar los resultados. Sin embargo, este proceso no es tan simple como se uno podría pensar. En particular, el modelo SST presenta un alto costo computacional, lo que significa que lograr la convergencia de una simulación desde cero hasta obtener una solución puede llevar un tiempo considerable y requerir una cantidad significativa de recursos computacionales.

Afortunadamente, TAU-Code ofrece la ventaja de permitir la inclusión de soluciones previas como datos iniciales para un nuevo estudio. Esta característica ayudará a ahorrar tiempo y a realizar un trabajo más estructurado. Dado que no se puede garantizar que el

código esté correctamente desarrollado hasta el momento que la simulación finaliza, esperar durante días o semanas por una solución que resulte ser incorrecta representaría una pérdida de tiempo considerable. Además, hay que considerar que muchas veces el desarrollo de un buen código, sobre todo en las primeras etapas, es un proceso iterativo de prueba y error para el programador, por lo que este necesita fluidez al correr el programa y obtener resultados, ya que es altamente probable que sea necesario repetir en múltiples ocasiones.

Según la experiencia previa, se sugiere llevar a cabo al menos dos estudios previos utilizando modelos menos costosos, pero menos precisos, para luego utilizar estos resultados como datos de entrada sucesivamente. Es decir, los resultados de la primera simulación como entrada de la segunda; y los resultados de la segunda como entrada de la simulación final.

El modelo SST en cuestión es un modelo RANS de dos ecuaciones. Esto implica que utiliza dos ecuaciones de transporte para describir la turbulencia en el flujo, lo cual proporciona una mayor precisión, pero, lógicamente, también implica un mayor costo computacional.

Por lo tanto, se recomienda realizar las dos primeras simulaciones utilizando un modelo de una ecuación, el cual tiene un menor costo computacional. Posteriormente, se puede pasar al modelo SST para obtener resultados más precisos, aunque a expensas de un mayor costo computacional.

En el anexo 2 se encuentran presentes todos los códigos completos utilizados, en la presente sección solo se incluirán detalles y pequeñas partes que poseen algún interés especial que desea ser resaltado.

Primera simulación

Por lo tanto, comenzando con la primera simulación, se tratarán de identificar detalles de interés, mostrar como fue llevada a cabo la misma, el tiempo requerido y los resultados obtenidos.

Desarrollos previos utilizan un modelo RANS de una ecuación llamado SST1, el mismo corresponde a una versión simplificada del modelo que actualmente se está estudiando con simplicidades en el cálculo de turbulencias.

Otra consideración a tener en cuenta es que se utilizara métodos de resolución de ecuaciones no lineales poco costoso, en este caso Backward-Euler. Siendo esta primera simulación solo una aproximación inicial y sirviendo para observar el correcto funcionamiento del código, se prioriza la rapidez y versatilidad antes que la precisión.

En tal sentido, se decide inicialmente por realizar una simulación de **5000 iteraciones**. La misma tomo aproximadamente **5 minutos** de computo, claramente entre los estándares buscados.

El primer resultado que se obtiene, es el presente en la siguiente gráfica. A esto se lo conoce como una traza y son ciertos parámetros que no tienen un significado específico físicamente, pero al observar el desarrollo del mismo, se puede tener una idea de si el programa convergió o se encuentra aun oscilando.

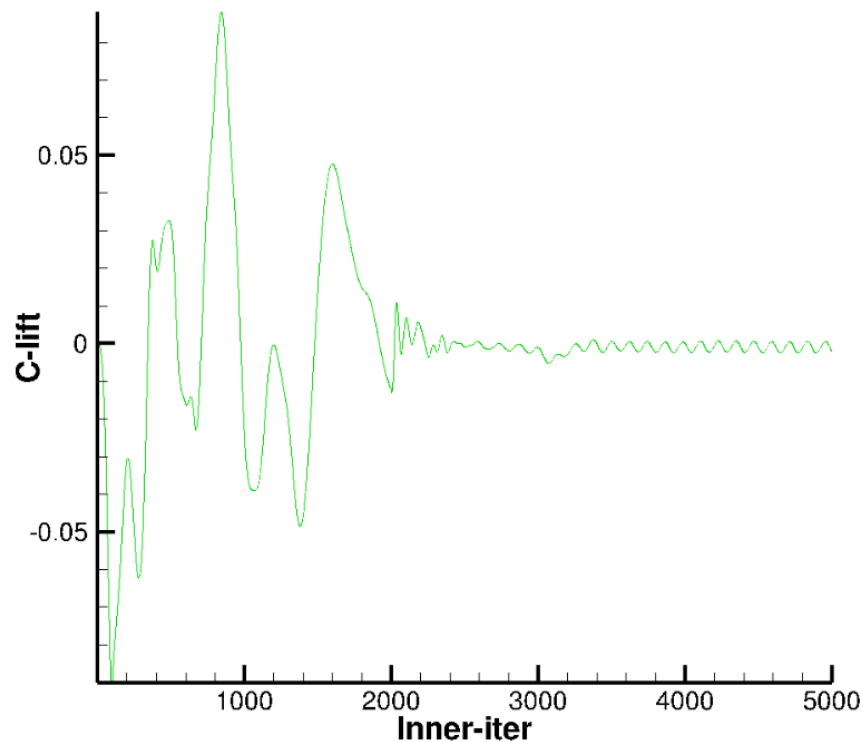


Figura 4-2: Traza C-lift para primera simulación:

Inesperadamente, el estudio parece encontrarse cerca de la estabilidad con solo 5000 iteraciones. De todas formas, esto tampoco significa que la solución es correcta ya que arribar a una buena solución, no depende únicamente de la convergencia, sino también de todos los demás factores y variables descriptas en el código.

Los resultados se pueden observar principalmente de dos formas distintas. Por un lado, gráficamente con escalas cromáticas en el modelo y luego extrapolando la información y realizando las mismas graficas que se vienen incluyendo a lo largo del desarrollo.

Una aclaración necesaria, es que los resultados de coeficientes de presión y rozamiento son solo sobre la superficie inferior del canal, por lo que no fue posible realizar una escala cromática, al menos con el software utilizado.

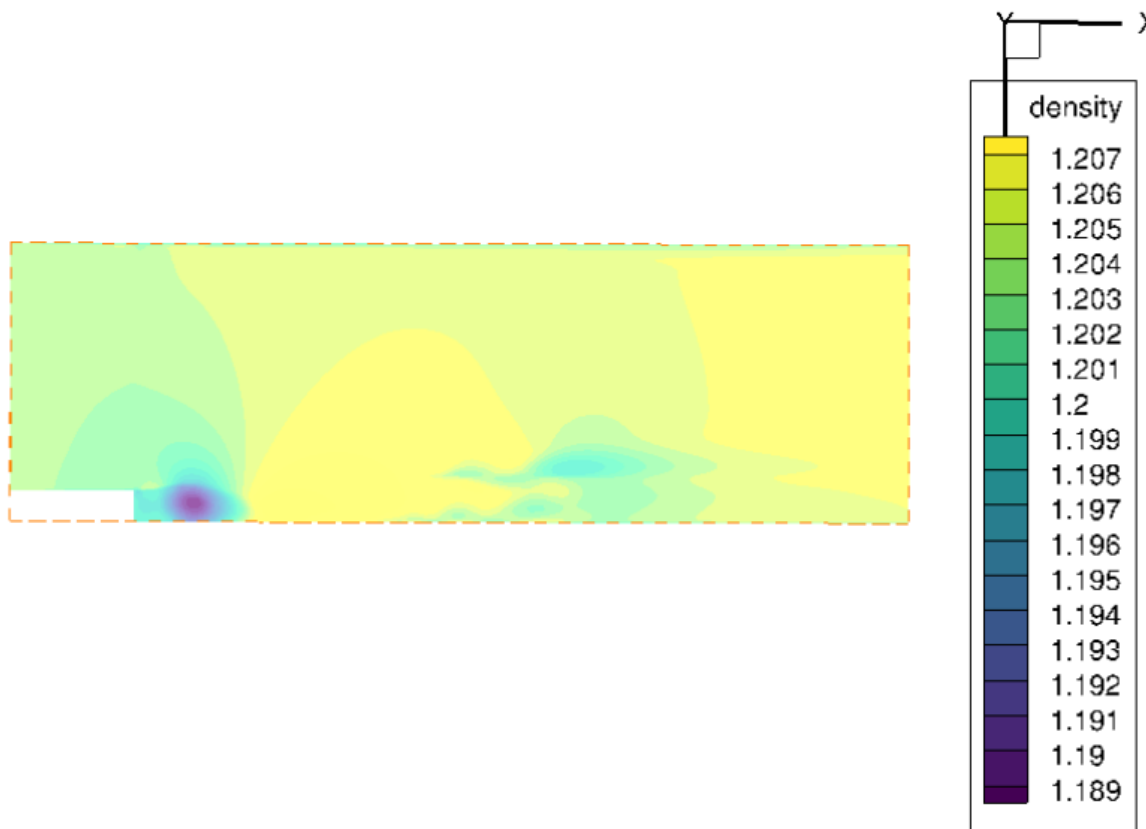


Figura 4-3: Densidad observada en modelo para primera simulación:

Las figuras 3-2 y 3-3 muestran los resultados de densidad y velocidad. Aunque la densidad no haya sido un valor a calibrar, es interesante de observar que las zonas de menor densidad corresponden a aquellas de mayor velocidad.

Los resultados a través de escalas cromáticas en el modelo no otorgan información al detalle sobre valores específicos, pero sirven como una excelente alternativa para observar que fenómeno está ocurriendo en cada lugar.

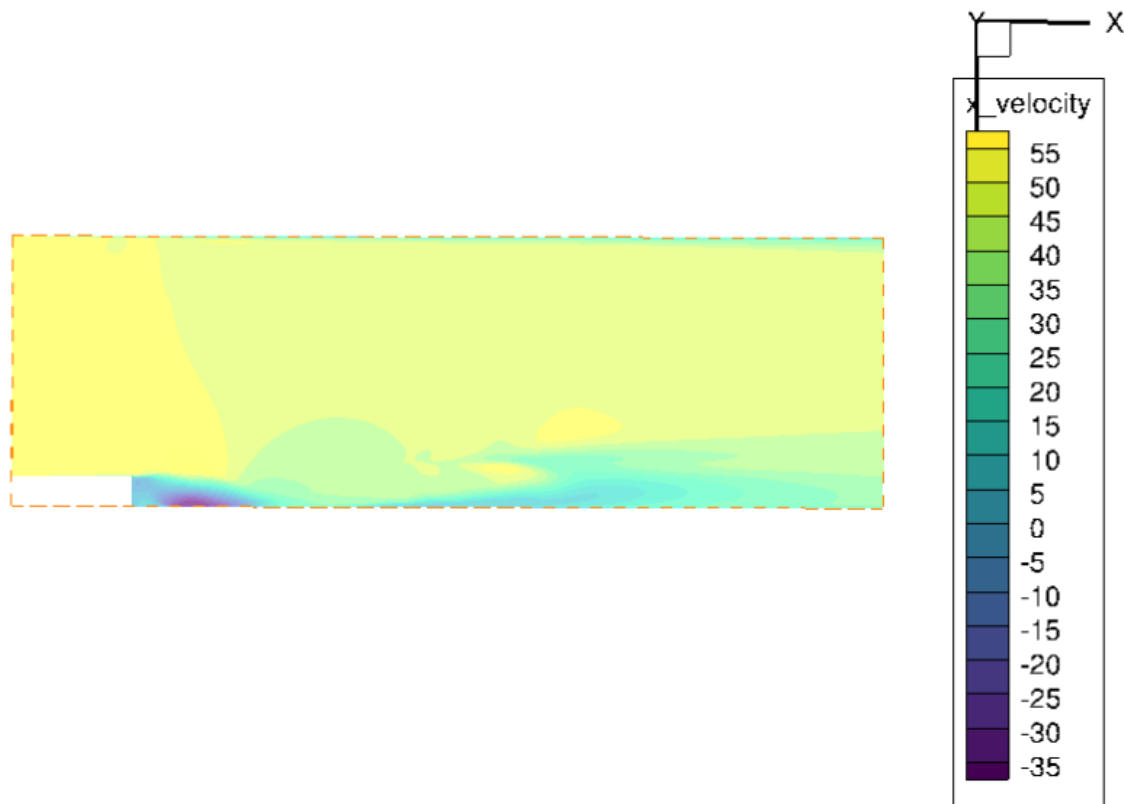


Figura 4-4: Velocidad observada en modelo para primera simulación:

En tanto, para obtener información detallada sobre valores específicos, se cuenta con la posibilidad de extrapolar datos y realizar las siguientes gráficas. **Las curvas en puntos corresponden a la información real, la que se encuentra en línea interrumpida a las simulaciones de la NASA y las curvas de línea continua a la simulación realizada.**

En la figura 3-4, se observan los datos correspondientes a perfiles de velocidades para las distintas posiciones de x . Los **resultados** son mucho **mejor a lo esperado** inicialmente para una simulación rápida. Por un lado, es algo de esperar al observar que

el estudio se encontraba cerca de la convergencia. El objetivo es que los resultados coincidan con las simulaciones de la NASA y si bien no lo cumple perfectamente, se considera al proceso como una gran primera aproximación.

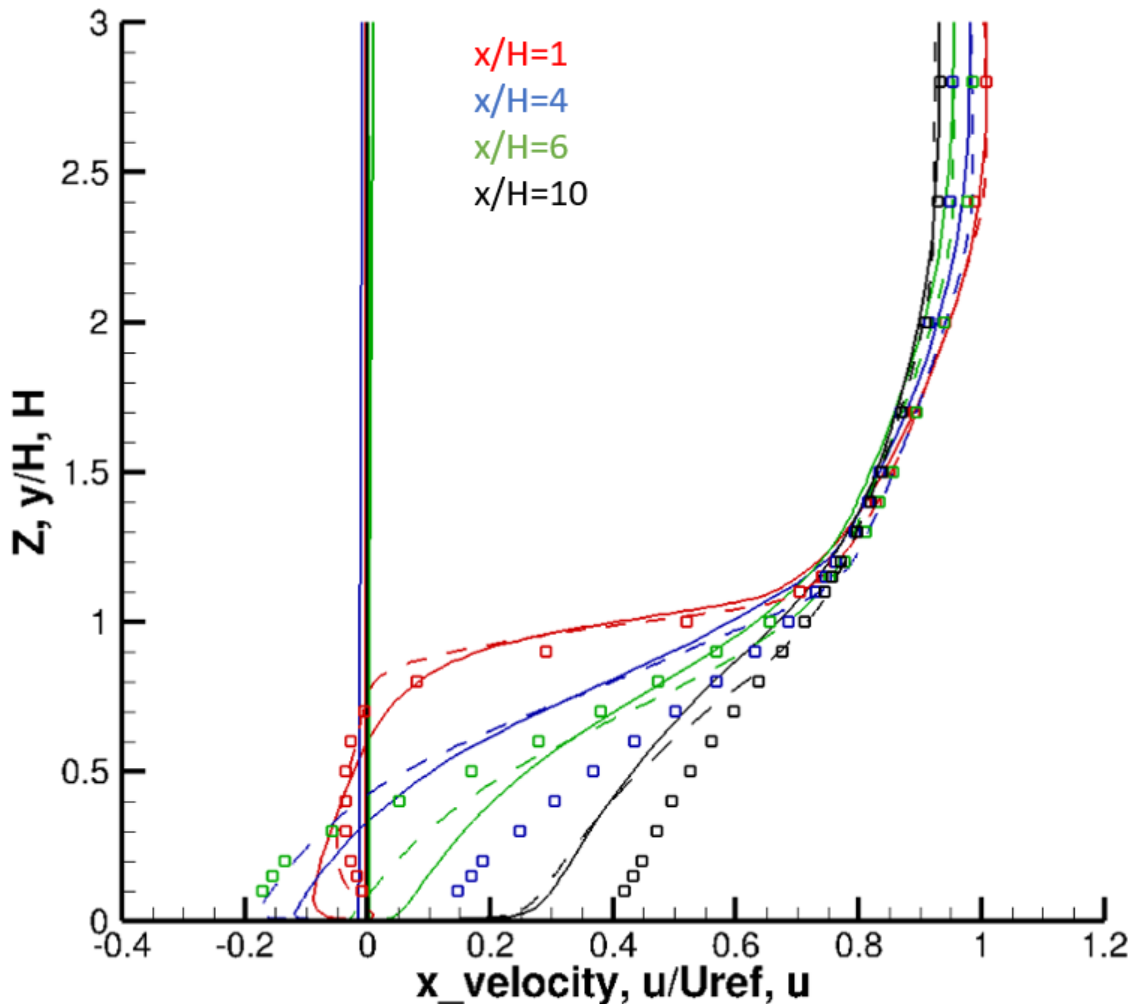


Figura 4-5: Perfiles de velocidad obtenidos en primera simulación:

En la figura 3-6, se encuentran los resultados extrapolados para valores de coeficientes de presión y rozamiento. Los resultados son igualmente de sorprendentes que para los perfiles de velocidad. Ocurre exactamente lo mismo, si bien, no es aun el resultado esperado, está a primera simulación es una muy buena primera aproximación.

Se debe aclarar que todas las gráficas presentaran líneas verticales, las mismas son una falsa representación y deberán ser no tenidas en cuenta, ya que no representan nada.

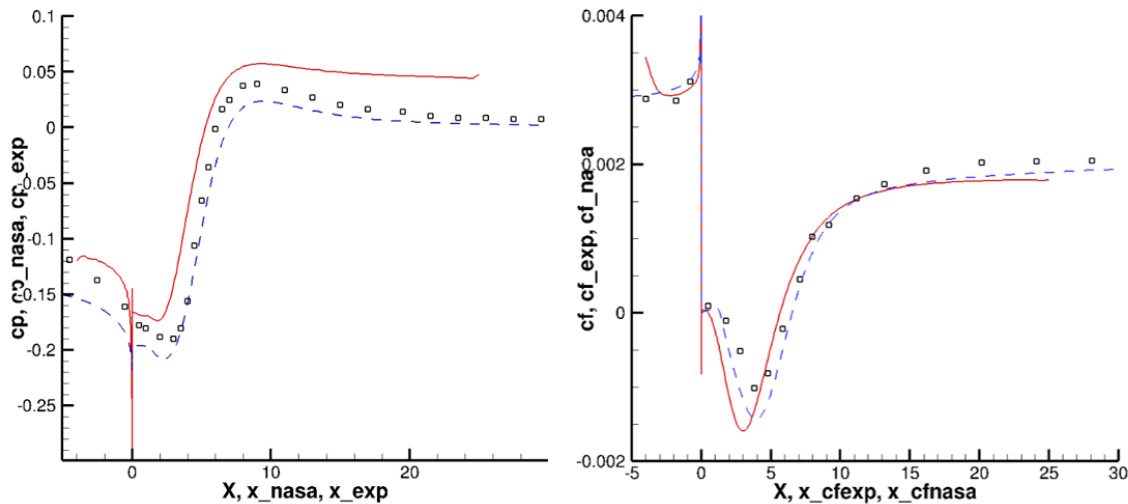


Figura 4-6: C_p y C_f obtenidos en primera simulación:

Como conclusión de la primera simulación, se podría decir que la misma resulto efectiva y será una gran primera aproximación al resultado buscado. Además, hay que resaltar que con una duración de aproximadamente 5 minutos y solo 5000 interacciones, se obtuvo una precisión acorde y mejor a la esperada inicialmente.

Segunda simulación

Viendo los resultados obtenidos en la primera simulación se cambió ligeramente el enfoque inicial. En un primer momento se pensaba en cambiar el modelo STT1 por otro más preciso y ya realizar una simulación de mediana complejidad.

Al encontrarse en la primera simulación con resultados ya buenos, se decidió por seguir utilizando el mismo modelo, solo con algunas pequeñas modificaciones en el código, como por ejemplo cambiar el método de resolución de ecuaciones no lineales de Backward-Euler por uno mas preciso como lo es Runge-Kutta 45. También en este caso se incluyo a las tensiones de Reynolds como resultado de salida a la simulación, lastimosamente esto fue olvidado en la primera simulación.

En este caso se **espera obtener resultados similares a los anteriores**, tal vez un poco mas precisos. Sería deseable que las pequeñas oscilaciones observadas en la gráfica de C_p (observable en la figura 3-6) desaparezcan. Sin embargo, más allá de lo nombrado,

esta simulación busca aligerar esfuerzos y acortar el tiempo de la simulación final, por lo que toda mejora es bienvenida, aun así, si no lo hubiera, con los resultados de la primera simulación, ya se considera suficiente para correr la simulación final.

Si observamos la traza de esta simulación, al igual que en la anterior, la misma se encuentra con pequeñas oscilaciones finales, lo que indica que está cerca de convergir.

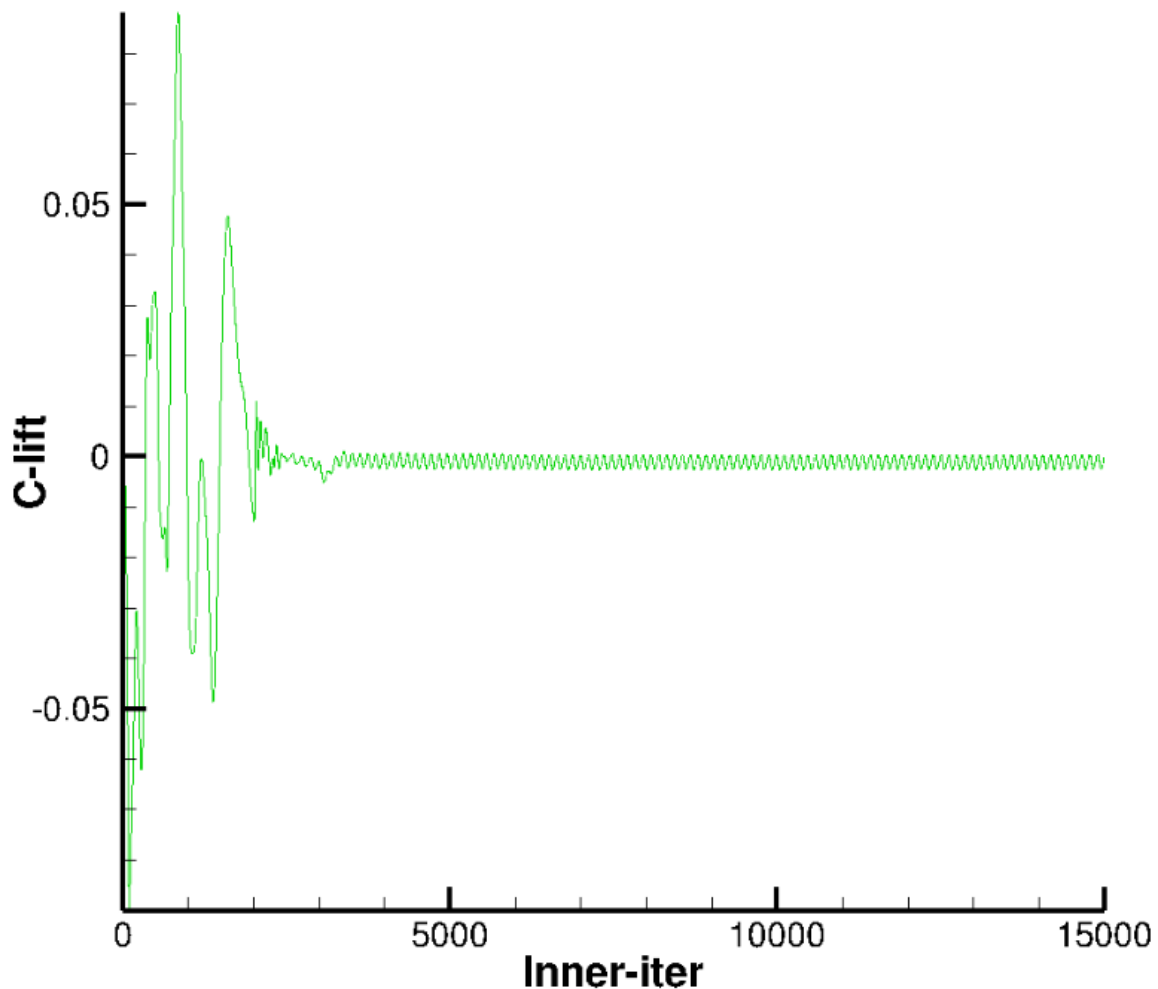


Figura 4-7: Trazas C-lift para segunda simulación:

Se optó por realizar **15.000 iteraciones** con un tiempo demandado aproximadamente de **15 minutos**. Los resultados se pueden observar a continuación:

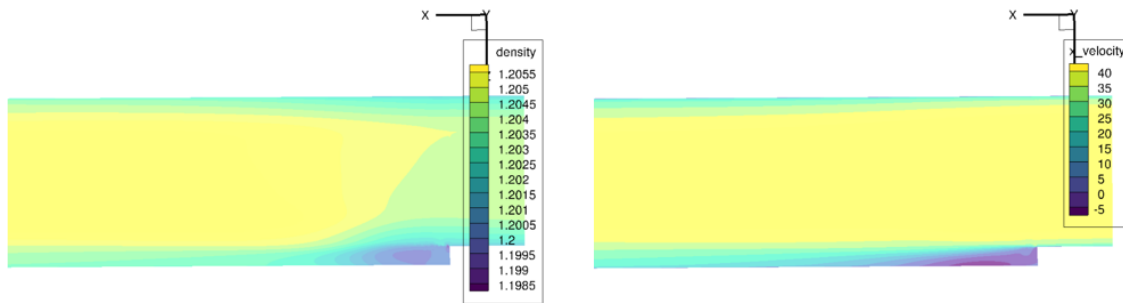


Figura 4-8: Densidad y Velocidad en X para segunda simulación:

Comparado con los resultados de la simulación anterior, se puede ver a los sectores de menor velocidad mas distribuidos por debajo del escalón, con una mayor longitud. Debido a esto se esperan mejores resultados, ya que si observamos las graficas de velocidad de la primera simulación en la figura 3-5, es fácil notar que los resultados de velocidad de la simulación eran algo mayores que los reales. Por lo que, al ver aquí, que hay una mayor área con valores de velocidad menor, da buenas expectativas frente a los resultados.

En este caso, también fue posible obtener una gráfica cromática de valores de tensiones de Reynolds, observable en la figura 3-9:

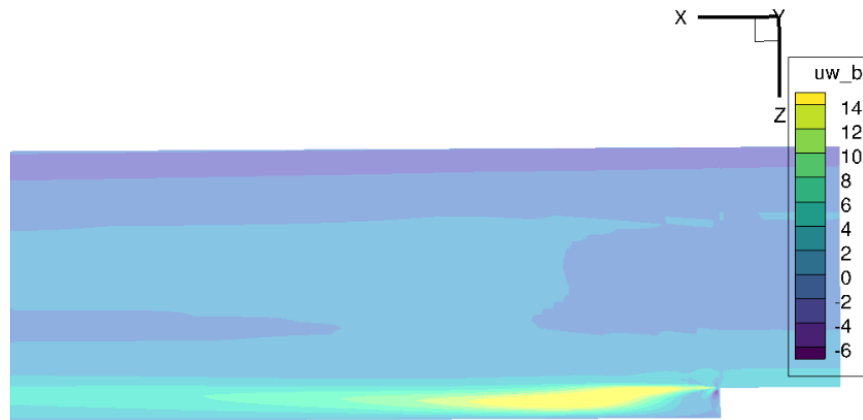


Figura 4-9: Tensiones de Reynolds en modelo:

Realizar conclusiones a partir de la anterior figura, no sería sencillo por lo que se aguardará a las graficas para poder comentar sobre lo mismo. Se podría mencionar que los resultados parecen ser mayores bajo el escalón, la zona en la que se esperaba esto.

En la figura 3-10 a continuación, se observan los resultados obtenidos para perfiles de velocidad. Como era de esperar, se observa un mejor ajuste que en la simulación anterior. Los resultados ya son muy buenos y con solo 15 minutos de simulación se obtienen valores ya comparables con los otorgados por el modelo SST utilizado por la NASA, un modelo de muchísimo mayor costo y que necesita de todo el trabajo de postproceso que requiere su posterior calibración.

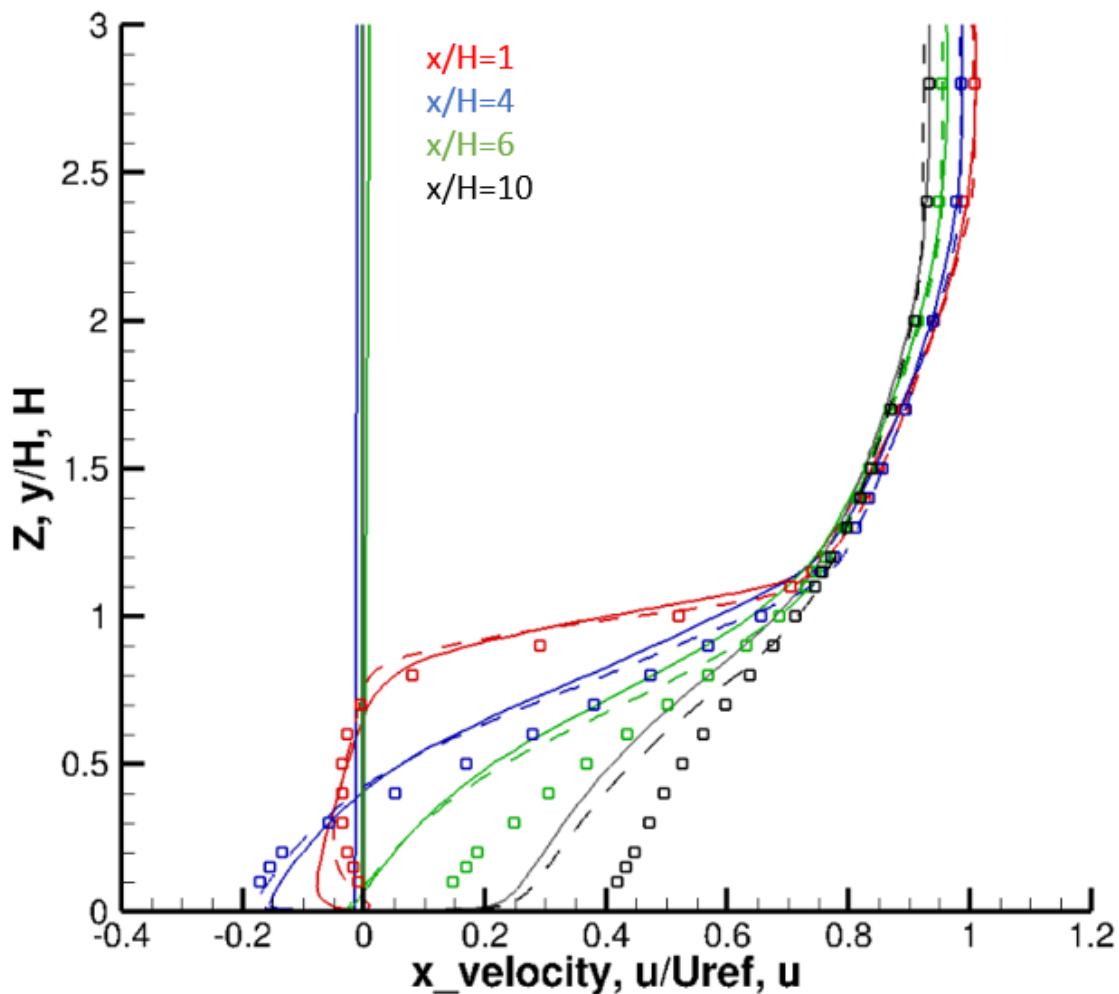


Figura 4-10: Perfiles de velocidad obtenidos en segunda simulación:

Desafortunadamente en cuanto a los resultados obtenidos para el coeficiente de presión, se puede considerar que hubo una desmejora respecto a los obtenidos en la anterior simulación. Los valores se alejaron algo más de los resultados experimentales y la cantidad de oscilaciones aumentó.

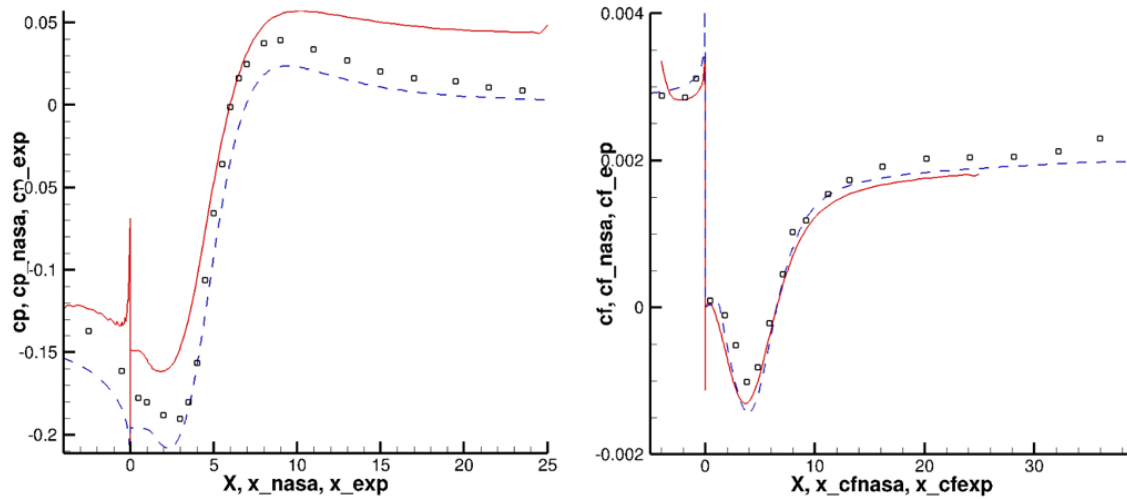


Figura 4-11: C_p y C_f obtenidos en segunda simulación:

No así ocurre con el coeficiente de rozamiento, el cual mejora y prácticamente ya corresponde al mismo resultado que el obtenido por la NASA. Es más, en este punto se podría decir que el resultado es tan bueno o mejor que el respaldado.

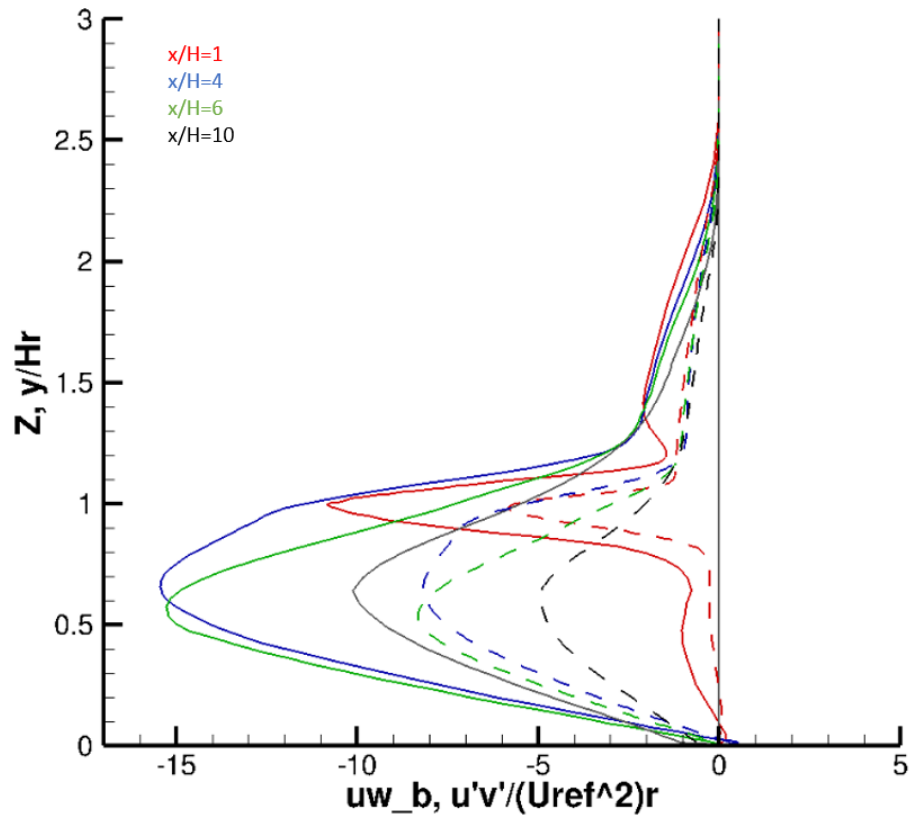


Figura 4-12: Tensiones de Reynolds obtenidas en segunda simulación:

Continuando con los resultados obtenidos, se muestra el correspondiente a tensiones de Reynolds. Recordando que los resultados en línea continua corresponden a los obtenidos en la simulación y los mostrados en línea a trazos a la simulación de la NASA. Se observa claramente que ante dicho resultado, la simulación otorga valores alejados de la realidad.

En resumen, de la segunda simulación se puede concluir que la misma presenta leves mejoras respecto a la primera. La convergencia no parece haber mejorado mucho, mientras que los valores de perfiles de velocidad y coeficiente de rozamiento se han ajustado mejor, esto se contrarresta con el empeoramiento en coeficiente de presión. A pesar de haber tenido resultados erróneos, es interesante haber sumado valores de tensiones de Reynolds.

Simulación final calibrada

Ya teniendo los dos resultados de las simulaciones iniciales, siendo esta una gran aproximación al resultado final buscado. Se encuentran las condiciones dadas para poder llevar a cabo la simulación final, ya si esta vez utilizando el modelo SST con los parámetros calibrados obtenidos previamente.

Si hubiera sido el caso que las simulaciones previas no hayan arribado a valores cercanos y si se creería necesario llevar a cabo otra de mayor numero de iteraciones, o bien utilizando un modelo computacional de complejidad media entre SST1 y SST. No hay nada que impida esto, pero dado los buenos valores mostrados en las simulaciones previas, esto no se consideró necesario.

Comenzando por la simulación lo primero a realizar, es tomar aquel código con el cual las 300 simulaciones de recolección de datos inicial fueron realizadas y modificar dos puntos principalmente. Por un lado, colocar como valores de entrada, los resultados de salida de la segunda simulación. Por el otro, cambiar los parámetros por los valores calibrados obtenidos.

Lo segundo es posible de observar en la figura 3-12, donde se observa un pequeño apartado del código y los valores utilizados, correspondientes los obtenidos en el desarrollo anterior.

```

Turbulence -----: -
Turbulence model version: k-w #RSM
Vortical flow correction (0/1): 1
Vortical flow correction model: sarc
SARC vortical correction coefficients: 1.2 8.6 1.4 1.0
SAS correction (0/1): 1
SAS Coefficient: 0.17 ##change ze
pressure gradient correction coefficients: 0
v2pkd correction coefficient: 6 ##change zero
k-w models -----: -
k-w model version: Menter_SST.1994

```

Figura 4-13 Cambio de parámetros en el código:

Para correr el programa, al ser la simulación final y no contar con mucho tiempo para prueba y error, se decidió por optar por llevar a cabo **240.000 iteraciones**. Dicho número no tiene un significado en especial, si no que en comparativa con las simulaciones anteriores es 16 veces mayor, lo que hacer pensar que ya se asegurará la convergencia.

Pensándolo en retrospectiva con las simulaciones anteriores, donde 15.000 iteraciones requirieron de cerca de 15 minutos, siendo 16 veces mayor esta vez, uno esperaría que toda la simulación necesite alrededor de 4 horas de computo.

Este caso no fue así, ya al tener un mayor costo computacional, cada iteración necesita de mas tiempo, haciendo que se necesiten **entre 48 y 72 horas**. Al cabo de dicho tiempo, la simulación se encontraba finalizada y solo restaba observar y comprender los resultados.

Al observar la traza, notamos grandes oscilaciones entre las iteraciones 50.000 y 100.000, esto llama la atención ya que podría representar grandes cambios de resultados entre aquellos puntos. Esto le da sentido al gran número de iteraciones por el que se optó.

Seguido se observa que ya a partir de la iteración número 150.000, el problema se encuentra totalmente estable sin presentar ninguna oscilación. Es posible interpretar como esto a una solución totalmente convergida y la final que daría el código de la manera que fue planteado el mismo. Por lo que los **resultados son definitivos** y comprobaran si la calibración fue exitosa o no.

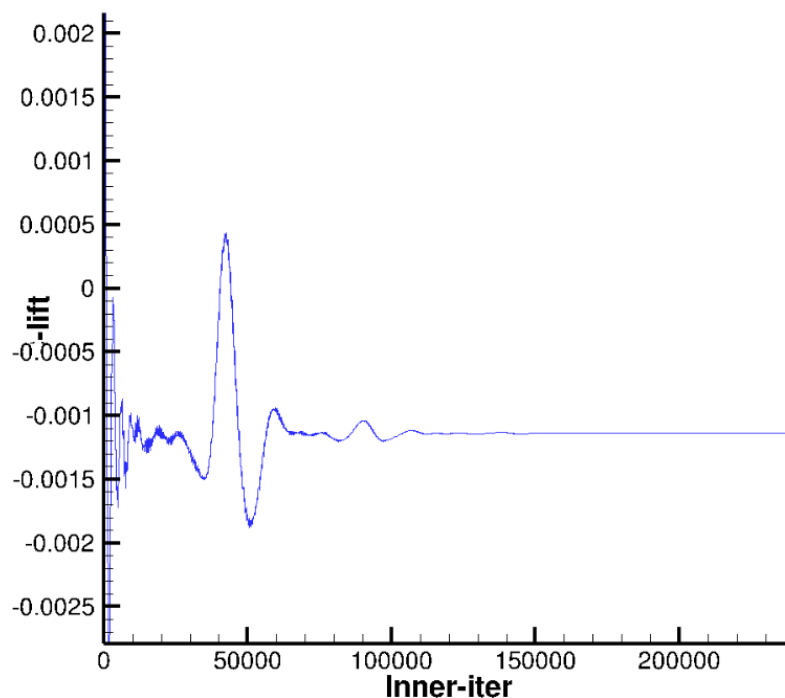


Figura 4-14: Trazas simulación final calibrada:

Continuando con los resultados. En un primer lugar se detallan los observables en los modelos con escalas cromáticas. Comenzando aquellos correspondientes a velocidad en x y densidad obsérvalos en la figura 3-14, la distribución es lógicamente la misma que para las imágenes anteriores. Lo positivo es que el grado de definición aumento considerablemente, se puede observar mas zonas de diferentes colores, lo que indicaría una mayor precisión en los resultados.

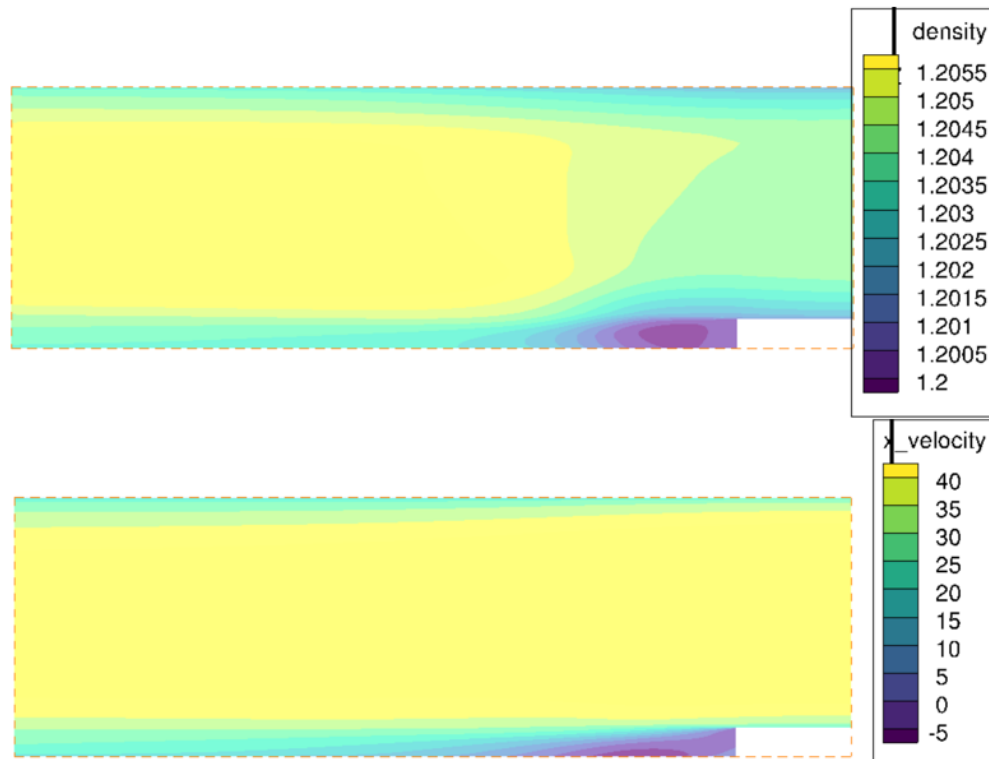


Figura 4-15: Densidad y Velocidad en X para simulación final:

Continuando por los resultados correspondientes a tensiones de Reynolds, estos son observables a continuación en la figura 3-14.

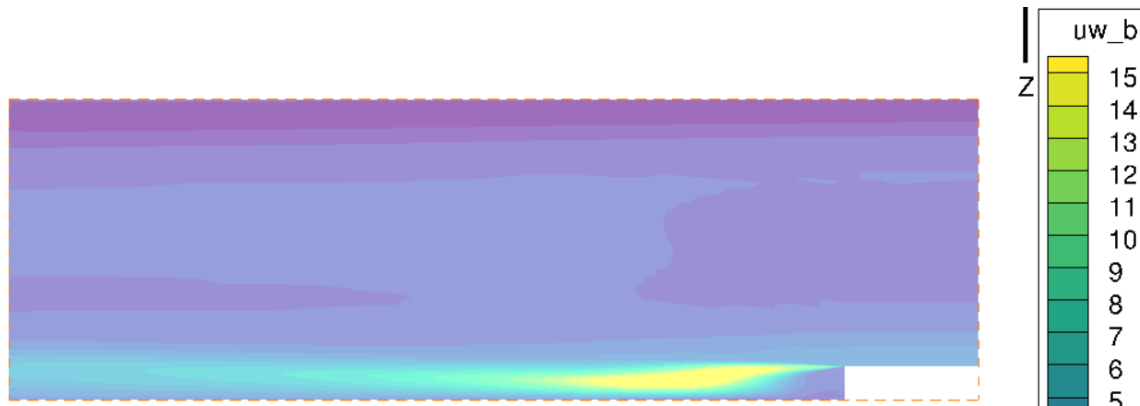


Figura 4-16: Densidad y Velocidad en X para simulación final:

Si bien es difícil interpretar resultados, se observan cambios respecto a la segunda ecuación y al ser estos erróneos, se podría esperar alguna mejora.

Observando las graficas de curvas de velocidad en x, las mismas ya coinciden prácticamente con las curvas obtenidas por la NASA en su simulación. Se observan las pequeñas mejoras necesarias con respecto a la simulación anterior, los resultados son satisfactorios para lo que se esperaba de la calibración.

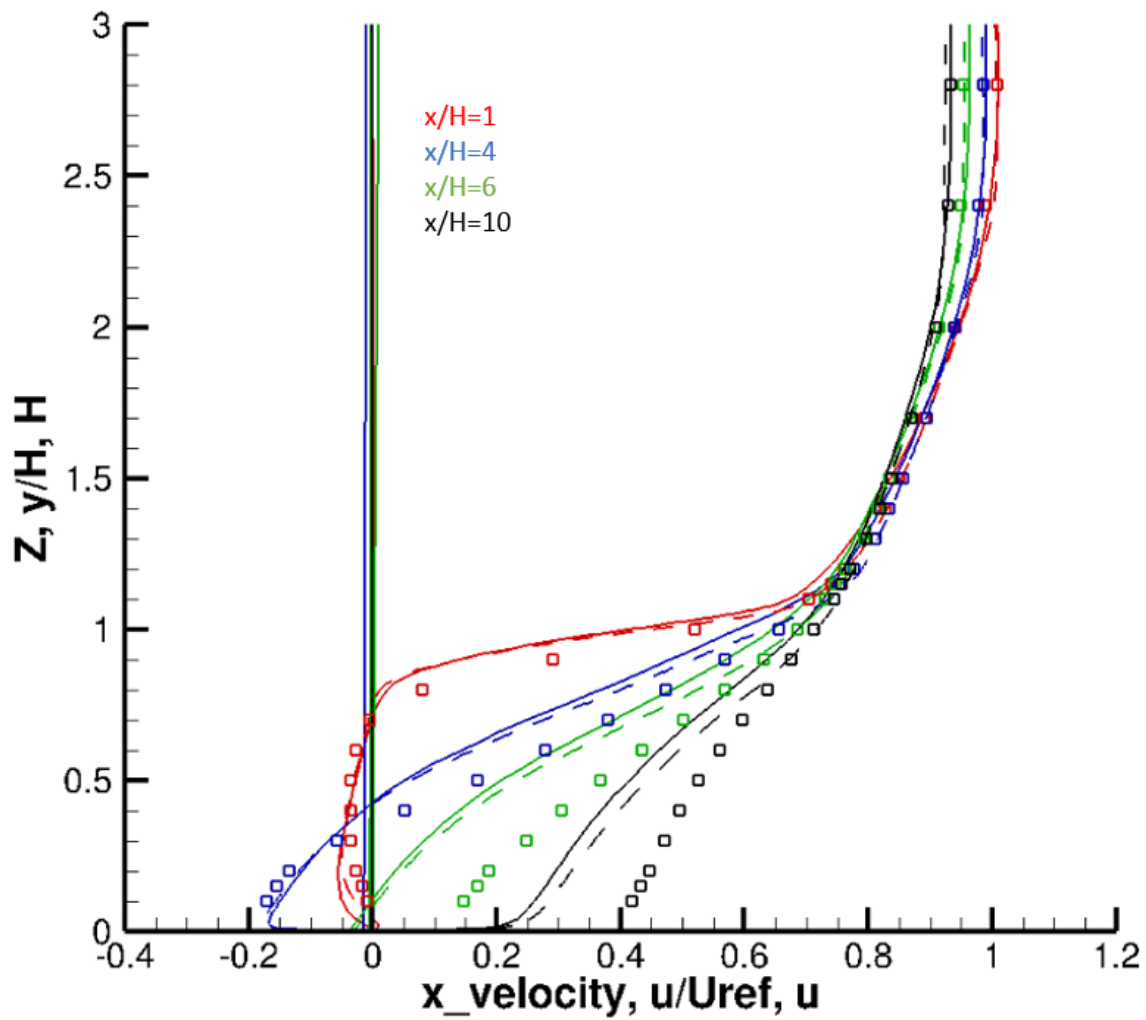


Figura 4-17: Perfiles de velocidad obtenidos en la simulación final calibrada:

Continuando con resultados para coeficientes de presión y rozamiento, podemos observar lo siguiente.

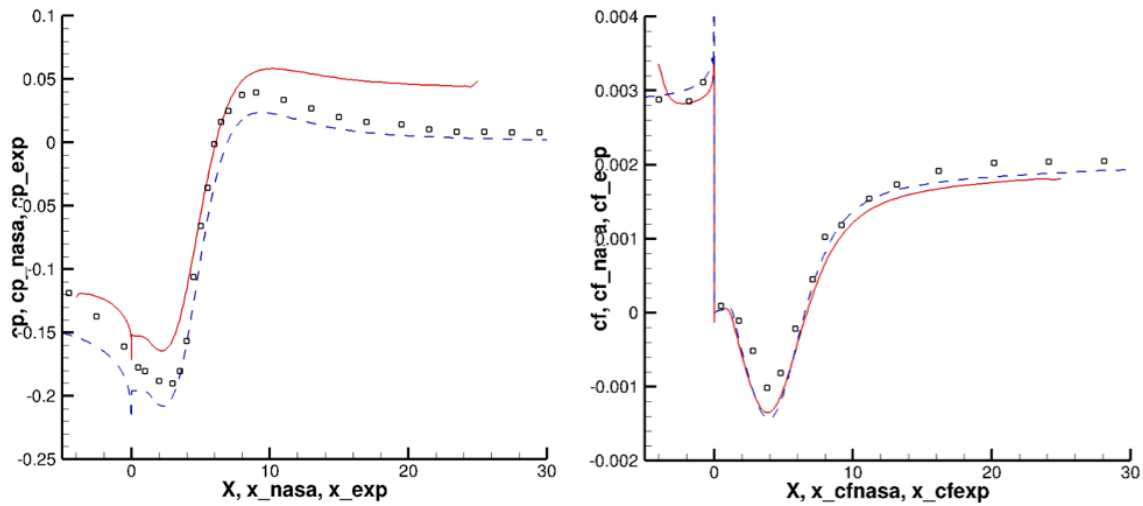


Figura 4-18: Cp y Cf obtenidos en simulación final calibrada:

En primer lugar, los valores de coeficiente de rozamiento coinciden perfectamente con la simulación de la NASA y por partes mejor que esta con los valores reales. Por lo tanto, en este caso, se puede considerar a la calibración mas que exitosa.

Respecto al resultado obtenido para coeficiente de rozamiento, el mismo no coincide con ninguno de los dos valores a comparar. No obstante, mientras que el simulador de la NASA otorga valores menores que los reales, la simulación realizada otorga valores ligeramente mayores. Lógicamente esto corresponde a un error, pero aun así se encuentra dentro de los márgenes aceptables y da un marco de comparación para futuros estudios. En tal sentido, se puede considerar a la simulación como calibrada.

Continuando por el último valor en estudio, se presenta a continuación las graficas correspondientes a tensiones de Reynolds en la figura 3-18. Los resultados prácticamente no mejoran respecto a la última simulación. Estos se consideran erróneos y no es posible arribar a mayores conclusiones a través de lo mismo.

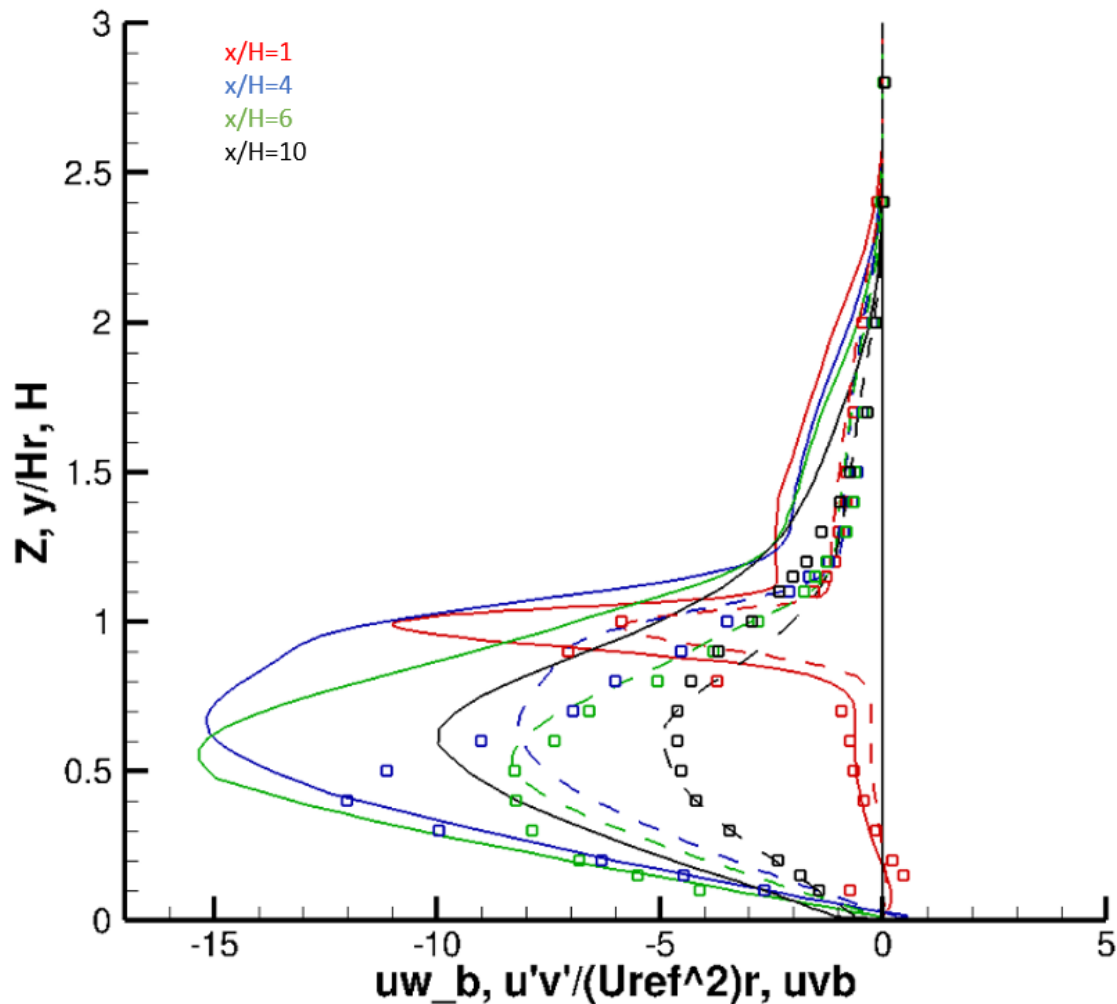


Figura 4-19: Tensiones de Reynolds obtenidas en simulación final calibrada:

Resumiendo, es posible decir que la simulación demostró una **correcta calibración**. Principalmente en resultados de velocidad en la dirección x y coeficientes de rozamiento, obteniendo resultados perfectamente calibrados, demostrando que el modelo funciona tan bien como el de la NASA para dichos valores. En cuanto al comportamiento del modulo respecto al coeficiente de presión, el mismo no se comporta como el de la NASA, pero aun así con su diferencia otorga valores representativos y los cuales podrán utilizarse para cualquier estudio lógico. Respecto a los resultados en tensiones de Reynolds, como era de esperar y se conocía de un principio, el modulo de simulaciones SST de TAU-Code no funciona correctamente y otorga resultados erróneos. No es posible interpretar estos resultados y llegar a conclusiones a través de estos.

Conclusiones y recomendaciones

Conclusiones

De hecho, las principales conclusiones se fueron detallando a lo largo del informe. Sin embargo, se obtuvieron resultados interesantes que serían importantes de resaltar, lo cual se realizará en el presente apartado.

En primer lugar y como fue detallado anteriormente se puede concluir que la **calibración fue exitosa**. Dentro de las limitaciones de una primera experiencia en este tipo de trabajos, se pudo llegar a resultados acordes.

Se obtuvieron en dos de los cuatro valores a calibrar (velocidad en dirección x y coeficiente de rozamiento) resultados prácticamente perfectos a lo máximo que uno puede aspirar en este caso, que es igualar a la NASA.

En tanto al coeficiente de presión, se podría decir que el programa no otorga los mismos resultados que la NASA pero que el valor obtenido se encuentra en un rango lógico y es un resultado representante del fenómeno.

En estos tres casos, los resultados obtenidos deben ser considerados representativos, pero no como verídicos, ya que como fue observado, los mismos no se corresponden perfectamente a los valores experimentales.

Respecto a valores de tensiones de Reynolds, los resultados son completamente erróneos. Si bien es posible observar, que la simulación realizada por la NASA ya otorga valores con un alto grado de error, los obtenidos en este caso son aun peores y no se considera que en este caso se pueda extrapolar a valores con los cuales hacer un análisis eficaz.

Siendo nombrado todo lo anterior, se concluye que el **modelo SST funciona correctamente bajo los primeros tres resultados** y que es posible **continuar utilizando el mismo** en futuros estudios.

Se debe tener en cuenta que es necesario determinar **cuándo vale la pena utilizar el modelo SST**. Siendo que su utilización **demanda una calibración de parámetros**, como

se realizó en este informe. Al considerar que simulaciones más simples, como las de los casos uno y dos, ofrecieron resultados muy similares y dada la gran cantidad de tiempo y esfuerzo requeridos para realizar la calibración, uno debe cuestionarse la necesidad de utilizar este modelo.

Recomendaciones

Las recomendaciones a afrontar un estudio de este tipo son varias. La experiencia obtenida al realizar este trabajo, siendo el mismo un camino lleno de obstáculos sorteados y un gran camino de aprendizaje, hace que uno tenga una mejor capacidad frente a la resolución de problemas similares en el futuro.

En primer lugar, la conclusión no es directamente del trabajo realizado, pero es para la persona que necesite hacer un estudio CFD. Considero que lo ideal antes de decidirse a utilizar las ecuaciones SST, se debería comenzar realizando una simulación mas simple sin calibración y observar los resultados.

Viendo que en las primeras simulaciones se obtuvieron resultados similares, tal vez en muchos casos llegar a realizar una simulación SST no sea necesario. Si fuera que la primera simulación no resulte con la precisión requerida, esta ya bien se podría utilizar como datos de entrada al realizar SST. Siendo así un ahorro muy significativo de tiempo y esfuerzos.

Por otro lado, a aquella persona que desee utilizar UQLab, le recomendaría comenzar afrontando casos sencillos para familiarizarse con el software y luego afrontar el caso final. Si bien el programa tiene una potencia de resolución muy interesante y resuelve en este caso, una calibración que de otra manera no podría ser computada. La dificultad a la hora de desarrollar el código no es menor y lamentablemente no es fácil encontrar respuestas y soluciones a los problemas que se presenten.

Otro cambio interesante a futuro es para la obtención del paquete de datos iniciales. Para dichas simulaciones sería importante conocer de antemano con que información experimental se contara para comparar, ya que si no ocurre como en este caso que se debe descartar una gran parte de los datos porque los mismos se encuentran para una posición que no se cuenta con valores de referencia.

Bibliografía

[1] Bergerson, J., & Muehleisen, R. T. (2015). Bayesian Large Model Calibration Using Simulation and Measured Data for Improved Predictions. SAE Int. J. Passeng. Cars - Mech. Syst., 8(2). doi:10.4271/2015-01-0481

[2] Kennedy, M. C., & O'Hagan, A. (2001). Bayesian Calibration of Computer Models. Journal of the Royal Statistical Society: Series B, 63(3), 425-464.

[3] Riddle, M., & Muehleisen, R. T. (2014). A Guide to Bayesian Calibration of Building Energy Models. Building Simulation Conference. Atlanta, GA: ASHRAE.

[4] P.-R. Wagner, J. Nagel, S. Marelli, B. Sudret, UQLab user manual – Bayesian inference for model calibration and inverse problems, Report UQLab-V2.0-113, Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland, 2022

[5] Ejemplo de uso de Teorema de Bayes, Disponible en: <https://www.fhybea.com/teorema-bayes.html>, Accedido: 03/03/2023.

[6] Métodos generales de estimación, Disponible en: http://www5.uva.es/estadmed/inferen/estima_punt/t9metod_ver.htm, Accedido: 01/07/2023

[7] Método máximo verosimilitud, Disponible en: <https://www.uv.es/ceaces/tex1t/4%20estimacion/emv.htm>, Accedido: 03/07/2023

[8] S. Marelli, N. Lüthen, B. Sudret, UQLab user manual – Polynomial chaos expansions, Report UQLab-V2.0-104, Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland, 2022

[9] UQLab: A Framework for Uncertainty Quantification in MATLAB, Stefano Marelli and Bruno Sudret, In The 2nd International Conference on Vulnerability and Risk Analysis and Management (ICVRAM 2014), University of Liverpool, United Kingdom, July 13-16, 2014, pp. 2554–2563. DOI: 10.1061/9780784413609.257

[10] TAU-Code User Guide, Institute of Aerodynamics and Flow Technology, July 10, 2018

[11] Technical Documentation of the DLR TAU-Code Release 2019.1.0, Institute of Aerodynamics and Flow Technologie Braunschweig / Gottingen, 2019

[12] Schwamborn, Dieter & Gerhold, Thomas & Heinrich, Ralf. (2006). The DLR TAU-code: Recent applications in research and industry.

[13] 2DBFS: 2D Backward Facing Step, NASA, Disponible en: https://turbmodels.larc.nasa.gov/backstep_val.html, Accedido: 13/07/2023

[14] J.H Ferziger & M Peric (2002). Rans Models, Computational Methods for Fluid Dynamics, ed. 03, pp. 292.

[15] Subbian, Botelho e Souza, Radespiel, Zander, Moshagen, Friedman, Moiola, Breitsamter & Sørensen. (2020). Calibration of an Extended Eddy Viscosity Turbulence Model Using Bayesian Update. AIAA JOURNAL. pp. 3-6. DOI: 10.2514/1.J060527

[16] Bishop, C. (2006). Bayes' theorem for Gaussian variables, Pattern Recognition and Machine Learning, pp. 90 -93.

Índice de figuras:

Figura 1-1: 2DBFS: 2D Backward Facing Step, Ilustración del problema [13]:	10
Figura 1-2: Resultados experimentales NASA [13]:	13
Figura 2-1: Combinación de parámetros en orden de uso:	16
Figura 2-2: Combinación de parámetros en orden ascendente:	16
Figura 2-3: Distribución inicial de perfiles de velocidad:	17
Figura 2-4: Distribución inicial de C_p y C_f :	18
Figura 2-5: Distribución inicial de tensiones de Reynolds:	19
Figura 2-6: Valores de referencia problema de ejemplo [5]:	21
Figura 3-1: Diagrama UQLab, entradas y salidas [9]:	28
Figura 3-2: Flujo de datos en UQLab [9]:	29
Figura 3-3: Archivo de combinaciones de parámetros:	30
Figura 3-4: Archivo ejemplo de combinación de resultados de simulación:	31
Figura 3-5: Arreglos finales en MatLab:	33
Figura 3-6: Distribución primaria:	34
Figura 3-7: Salida primaria de UQLAB:	37
Figura 3-8: Distribución posterior de parámetros calibrados:	38
Figura 3-9: PCE de velocidad evaluado en valores calibrados:	39
Figura 3-10: PCE de C_p y C_f evaluado en valores calibrados:	40
Figura 3-11: PCE de tensiones de Reynolds evaluado en valores calibrados:	41
Figura 4-1: Modelo mallado otorgado por la NASA [13]:	43
Figura 4-2: Traza C-lift para primera simulación:	46
Figura 4-3: Densidad observada en modelo para primera simulación:	47
Figura 4-4: Velocidad observada en modelo para primera simulación:	48
Figura 4-5: Perfiles de velocidad obtenidos en primera simulación:	49
Figura 4-6: C_p y C_f obtenidos en primera simulación:	50
Figura 4-7: Traza C-lift para segunda simulación:	51
Figura 4-8: Densidad y Velocidad en X para segunda simulación:	52
Figura 4-9: Tensiones de Reynolds en modelo:	52
Figura 4-10: Perfiles de velocidad obtenidos en segunda simulación:	53

Figura 4-11: Cp y Cf obtenidos en segunda simulación:	54
Figura 4-12: Tensiones de Reynolds obtenidas en segunda simulación:	55
Figura 4-13: Cambio de parámetros en el código:	56
Figura 4-14: Traza simulación final calibrada:	57
Figura 4-15: Densidad y Velocidad en X para simulación final:	58
Figura 4-16: Densidad y Velocidad en X para simulación final:	58
Figura 4-17: Perfiles de velocidad obtenidos en la simulación final calibrada:	59
Figura 4-18: Cp y Cf obtenidos en simulación final calibrada:	60
Figura 4-19: Tensiones de Reynolds obtenidas en simulación final calibrada:	61

Anexo 1: Código MatLab y Uqlab

Contents

- Bayesian inversion 2D Backward Facing Step
- Upload Data
- Calibration Steps
- Prior Data
- Surrogate Model
- Experimental Data
- BAYESIAN ANALYSIS
- Post analysis

Bayesian inversion 2D Backward Facing Step

Augusto Dovetta - Braunschweig, 31.03.2023

```
clearvars
rng(100,'twister')
uqlab
```

Unrecognized function or variable 'uqlab'.

Error in Backward_Facing_Step_2D_12_12_WORKS (line 5)
uqlab

Upload Data

```
files=dir('*.csv');
for i=1:length(files)
    DATA(i)=importdata(files(i).name); %CFD Data
end

for i=1:300
    DATA(i).data(1,:)=[];
    DATA(i).data(225,:)=[];
    DATA(i).data(449,:)=[]; %Deleting out of order lines
    DATA(i).data(896,:)=[];
end

%Simulation parameters

PriorData=readtable('prior_points/points.csv');
PriorDataSorted= sortrows(PriorData,6);
PriorDataSorted= table2array(PriorDataSorted(:,1:5));

%Experimental DATA

ExpData=readtable('exp_mod/exp_mod.csv');
ExpData= table2array(ExpData(:,1:6));

%Reorganise DATA
for i=1:300
    DveI(i,:)=DATA(i).data(:,3);
    DCp(i,:)=DATA(i).data(:,4);
    DUw_b(i,:)=DATA(i).data(:,5);
    DCF(i,:)=DATA(i).data(:,6);
end

%Take only the lower wall for Cp
i=1;
cont=1;
while (i <= width(DCp))
    if DATA(1).data(cont,2)~=0
        DCp(:,i)=[];
        i=i-1;
    end
    i=i+1;
    cont=cont+1;
end

%Repeat for CF
i=1;
cont=1;
while (i <= width(DCp))
    if DATA(1).data(cont,2)~=0
        DCF(:,i)=[];
        i=i-1;
    end
    i=i+1;
    cont=cont+1;
end
```

Calibration Steps

Prior Data

```
Inputopts.Inference.Data = PriorDataSorted;
myPriorDist = uq_createInput(Inputopts);
```

Surrogate Model

First for Velocity

```

MetaOpts1.Type = 'Metamodel';
MetaOpts1.MetaType = 'PCE';

MetaOpts1.ExpDesign.X = PriorDataSorted;
MetaOpts1.ExpDesign.Y = DVelX;

VelXModel = uq_createModel(MetaOpts1);

% For Cp
MetaOpts2.Type = 'Metamodel';
MetaOpts2.MetaType = 'PCE';

MetaOpts2.ExpDesign.X = PriorDataSorted;
MetaOpts2.ExpDesign.Y = DCp;

CPModel = uq_createModel(MetaOpts2);

% For Uw_b
MetaOpts3.Type = 'Metamodel';
MetaOpts3.MetaType = 'PCE';

MetaOpts3.ExpDesign.X = PriorDataSorted;
MetaOpts3.ExpDesign.Y = DUw_b;

Uw_BModel = uq_createModel(MetaOpts3);

% For Cf
MetaOpts4.Type = 'Metamodel';
MetaOpts4.MetaType = 'PCE';

MetaOpts4.ExpDesign.X = PriorDataSorted;
MetaOpts4.ExpDesign.Y = DCF;

CFModel = uq_createModel(MetaOpts4);

% All Models in one:
mySurrogateModel(1).Model = VelXModel;
mySurrogateModel(2).Model = CPModel;
mySurrogateModel(3).Model = Uw_BModel;
mySurrogateModel(4).Model = CFModel;

```

Experimental Data

```

myData(1).y = ExpData(1:112,3); %Exp Data Vel X
myData(2).y = ExpData(113:144,4); %Exp Data Cp
myData(3).y = [ExpData(1:47,5);ExpData(49:112,5)]; %Exp Data Uw_b --- Delete in 0
myData(4).y = ExpData(146:165,6);

myData(1).MOMap = [ones(1,width(myData(1).y)); ...
    1:width(myData(1).y)];

myData(2).MOMap = [2*ones(1,width(myData(2).y)); ...
    1:width(myData(2).y)];

myData(3).MOMap = [3*ones(1,width(myData(3).y)); ...
    1:width(myData(3).y)];

myData(4).MOMap = [4*ones(1,width(myData(4).y)); ...
    1:width(myData(4).y)];

```

BAYESIAN ANALYSIS

```

BayesOpts.Type = 'Inversion';
BayesOpts.Data = myData;
BayesOpts.Prior = myPriorDist;
BayesOpts.ForwardModel = mySurrogateModel;

myBayesianAnalysis = uq_createAnalysis(BayesOpts);

% Print out a report of the results:
uq_print(myBayesianAnalysis)
uq_display(myBayesianAnalysis)

```

Post analysis

```

%Graficar VelX
% SM=uq_evalModel([6 1.2 8.6 1.4 0.17],VelXModel),
SM=uq_evalModel([6 1.2 8.6 1.4 0.17],VelXModel); % Hay que correr el pce de VelX
plot(SM(1:224),DATA(1).data(1:224,2),'r',ExpData(1:28,2),'r+',SM(225:448),DATA(1).data(225:448,2),'b',ExpData(29:56,3),ExpData(29:56,2),'b*','s',
xlim([-0.4 1.5])
ylim([0 3])
legend('x/H=1','x/H=4','x/H=6','x/H=10','')
grid minor
xlabel('u/Uref')
ylabel('y/H')
axis square

% CP

```

```

SM=uq_evalModel([6 1.2 8.6 1.4 0.17],CPModel);
plot(DATA(1).data(897:1152,1),SM,'r',ExpData(113:145,1),ExpData(113:145,4),'r*')
xlim([-5 30])
ylim([-0.3 0.1])
grid minor
xlabel('x/H')
ylabel('Cp')
axis square

% Uw_b
SM=uq_evalModel([6 1.2 8.6 1.4 0.17],Uw_BModel);
plot(SM(1:224),DATA(1).data(1:224,2),'r',ExpData(1:28,5),ExpData(1:28,2),'r*',SM(225:448),DATA(1).data(225:448,2),'b',ExpData(29:56,5),ExpData(29:56,2),'b*',SI
ylim([0 3])
legend('x/H=1','','x/H=4','','x/H=6','','x/H=10','')
grid minor
xlabel('u\`v/Uref')
ylabel('y/H')
axis square

% Cf
SM=uq_evalModel([6 1.2 8.6 1.4 0.17],CfModel);
plot(DATA(1).data(897:1152,1),SM,'r',ExpData(146:165,1),ExpData(146:165,6),'r*')
xlim([-5 30])
ylim([-0.002 0.004])
grid minor
xlabel('x/H')
ylabel('Cf')
axis square

```

Contents

- [Only initial DATA to plot](#)

Only initial DATA to plot

Augusto Dovetta - Braunschweig, 31.03.2023

```
close all
% Upload Data
files=dir('*.csv');
for i=1:length(files)
    DATA(i)=importdata(files(i).name); %CFD Data
end

for i=1:300
    DATA(i).data(1,:)=[];
    DATA(i).data(225,:)=[];
    DATA(i).data(449,:)=[]; %Deleting out of order lines
    DATA(i).data(896,:)=[];
end

%Simulation parameters

PriorData=readtable('prior_points/points.csv');
PriorDataSorted= sortrows(PriorData,6);
PriorDataSorted= table2array(PriorDataSorted(:,1:5));

%Experimetal DATA

ExpData=readtable('exp_mod/exp_mod.csv');
ExpData= table2array(ExpData(:,1:6));

figure(1)
t = tiledlayout(2,2)
nexttile
for i=1:300
    hold on
    plot(DATA(i).data(1:224,3),DATA(i).data(1:224,2))
end
xlim([-0.4 1.5])
ylim([0 3])
grid on
axis square
hold off

nexttile
for i=1:300
    hold on
    plot(DATA(i).data(225:448,3),DATA(i).data(225:448,2))
end
xlim([-0.4 1.5])
ylim([0 3])
grid on
axis square
hold off

nexttile
for i=1:300
```

```

        hold on
        plot(DATA(i).data(449:672,3),DATA(i).data(449:672,2))
    end
    xlim([-0.4 1.5])
    ylim([0 3])
    grid on
    axis square
    hold off

    nexttile
    for i=1:300
        hold on
        plot(DATA(i).data(673:896,3),DATA(i).data(673:896,2))
    end
    xlim([-0.4 1.5])
    ylim([0 3])
    grid on
    axis square
    hold off

    t.Padding = 'compact';
    t.TileSpacing = 'compact';
    title(t,'Vel X Distribution')
    xlabel(t,'u/Uref')
    ylabel(t,'y/H')

    figure(2)
    t = tiledlayout(1,2)
    nexttile
    for i=1:300
        hold on
        plot(DATA(i).data(897:1152,1),DATA(i).data(897:1152,4))
    end
    xlim([-5 30])
    ylim([-0.3 0.1])
    grid minor
    xlabel('x/H')
    ylabel('Cp')
    axis square
    title('Cp Lower Wall')
    hold off

    nexttile
    for i=1:300
        hold on
        plot(DATA(i).data(897:1152,1),DATA(i).data(897:1152,6))
    end
    xlim([-5 30])
    ylim([-0.002 0.004])
    grid minor
    xlabel('x/H')
    ylabel('Cf')
    axis square
    title('Cf Lower Wall')
    hold off

    t.Padding = 'compact';
    t.TileSpacing = 'compact';

    figure(3)
    t = tiledlayout(2,2)
    nexttile
    for i=1:300
        hold on
        plot(DATA(i).data(1:224,5),DATA(i).data(1:224,2))

```

```

end
ylim([0 3])

grid minor
axis square
hold off

nexttile
for i=1:300
    hold on
    plot(DATA(i).data(225:448,5),DATA(i).data(225:448,2))
end
ylim([0 3])

grid minor
axis square
hold off

nexttile
for i=1:300
    hold on
    plot(DATA(i).data(449:672,5),DATA(i).data(449:672,2))
end
ylim([0 3])

grid minor
axis square
hold off

nexttile
for i=1:300
    hold on
    plot(DATA(i).data(673:896,5),DATA(i).data(673:896,2))
end
ylim([0 3])

grid minor
axis square
hold off
t.Padding = 'compact';
t.TileSpacing = 'compact';
title(t,'uv b Distribution')
xlabel(t,'u' b '/Uref2')
ylabel(t,'y/H')

x=1:300;
figure(11)
t = tiledlayout(5,1)
PriorDataSorted=sort(PriorDataSorted);
nexttile
plot(x,PriorDataSorted(:,1),'*r')
ylabel('Cw4')

nexttile
scatter(x,PriorDataSorted(:,2),'*b')
ylabel('Cr1')

nexttile
plot(x,PriorDataSorted(:,3),'*g')
ylabel('Cr2')

nexttile
scatter(x,PriorDataSorted(:,4),'*m')
ylabel('Cr3')

```

```
nexttile
plot(x,PriorDataSorted(:,5),'*k')
ylabel('ζ2')

title(t,'Combinaciones aleatorias de coeficientes')
% Move plots closer together
t.TileSpacing = 'compact';
```

Anexo 2: Código TAU-Code

BOUNDARY MAPPING

```

Markers: 1,6,11
Type : symmetry plane
Name: side1
# Write surface data (0/1): 0
block end

```

```

Markers: 2,7,12
Type : symmetry plane
Name: side2
# Write surface data (0/1): 0
block end

```

```

Markers: 5,9,13,14
Type: viscous wall
Subtype: turbulent
Write surface data (0/1): 1
Name: stepwall
block end

```

```

Markers: 4,10,15
Type: viscous wall
Subtype: turbulent
# Write surface data (0/1): 1
Name: ceiling
block end

```

```

Markers: 8
Type: farfield
Name: Inflow
Modify farfield file: mod-input.dat
# Write surface data (0/1): 1
block end

```

```

Markers: 3
Type: exit-pressure outflow
Name: Outflow
Match measured pressure (0/1): 0
Measured pressure: 101325
Measurement coordinates: -0.05 0.5 -0.0508
Matching iteration period: 2000
Matching iteration start: 1000
# Write surface data (0/1): 1
Exit pressure: 101495
block end

```

Cut plane output

Plane output description file: (thisfile)
 Plane output values: xyz_cp_cf_yplus_cfsgnx_cfxyz_edgeP0_edgeP1
 Plane output period: 5000
 Number of planes: 0
 Boundary type (euler/ns): ns
 Plane support x: 0.
 Plane support y: -0.05
 Plane support z: 0.
 Plane normal x: 0.
 Plane normal y: 1.
 Plane normal z: 0.

Surface output

Surface output description file: (thisfile)
 #Surface output files prefix: surface
 Surface output values: gidx_xyz_cp_cf_yplus_cfxyz_mean-cf_mean-cp
 # Surface output values: gidx_xyz_restart
 Surface output period: 5000

Extra field pointdata output

Field output description file: (thisfile)
 Field output values: cp_wdist_muettmue_visc_mean_wnormal_variance_rs-boussinesq

Solver parameters:

Files/IO -----: -
 Restart-data prefix: (none)
 Boundary mapping filename: (thisfile)
 Output files prefix: solution/SST1
 Profile output description file: (none)
 Automatic parameter update (0/1): 1
 Accumulate queue time (0/1): 1

Preprocessor parameters:

Files/IO -----: -
 Boundary mapping filename: (thisfile)
 Primary grid filename: ./bfs_ds_re38000_rans.grd
 Grid prefix: dualgrid/dualgrid
 Old grid prefix: (none)
 Parent input file prefix: (none)
 Parent output file prefix: (none)
 Output MG agglomeration (0/1): 0
 Type of discretisation (node=0/cell=1): 0
 : -
 Read partitioning filename: (none)
 Write graph filename: (none)

Parameter -----: -
 Number of multigrid levels: 4

```

Output level: 5
Chimera interpolation linear/nonlinear (0/1): 0
Runtime optimisation -----: -
  Cache-coloring (0/max_faces in color): 50000
  Bandwidth optimisation (0/1): 1
  2D offset vector (0 / x=1,y=2,z=3): 2
  Compute lugs mapping (0/1): 1
Agglomeration -----: -
  Point fusing reward: 1.2
  Structured grid coarsening: 0
Extras -----: -
  Sharp edge angle (degrees): 0
  Control volume edge weight (0/1): 1
  Change boundary control volumes (0/1): 0
  Compute DES scale(0/1): 0
  Compute exact whirlflux(0/1): 0
  Compute exact surface(0/1): 0
  Axisymmetry (0/1): 0
  Axisymmetry axial direction: 1 0 0
  Axisymmetry radial direction: 0 0 1
Partitioning -----: -
  Number of domains: 8
  Smooth partitions: 0
  Type of partitioning (name): private
  Metis parameter CoarsenTo (int): 100
  Metis parameter Mtype (int): 21
  Metis parameter Rtype (int): 13
  Metis parameter IType (int): 4
Theta not for Tau -----: -
  Preprocessing for incompressible solver (0/1): 0
  Correct metric for boundary control volumes (0/1): 0
  Project boundary control volumes coordinates (0/1): 0
  Translation factor for shifted boundary points: 1
  Compute parent faces (0/1): 0
-----
SOLVER
-----
Memory management -----: -
  Increase memory (0/1): 1
Variables -----: -
  Fix negative values (0/1): 1
Flux -----: -
  Inviscid flux discretization type: Central
Central flux -----: -
  Central dissipation scheme: Scalar_dissipation
  Central convective meanflow flux: Average_of_flux
  Scaling of central scheme dissipation: Standard
  Central convective turbulence flux: Roe
  2nd order dissipation coefficient: 0
  Inverse 4th order dissipation coefficient: 64
  Version of cell stretching coefficient: HyperFlex #TAU
  Use modified dissipation for 2D (0/1): 1 #0
Gradients -----: -
  Reconstruction of gradients: Green_Gauss #Least_square
Flowfield initialization -----: -

```

```

Init total conditions (0/1): 0
Relaxation -----: -
    Relaxation solver: Backward_Euler #Runge_Kutta#
Backward Euler -----: -
    Linear solver: Lusgs
    Linear preconditioner: (none)
    Implicit overrelaxation omega: 1
    Implicit overrelaxation beta: 1
LUSGS -----: -
    Sgs stages maximum: 3
    Lusgs increased parallel communication (0/1): 1
    Lusgs treat whirl implicitly (0/1): 0
Runge Kutta -----: -
    #Number Runge-Kutta steps: 3
    # Runge-Kutta coefficients: 0.666666666666667 0.666666666666666 1 0 0 0
Multigrid -----: -
    MG description filename: 3w++
    SG start up steps (fine grid): 2000
    MG source terms (0/1): 1
Full multigrid -----: -
    Multigrid start level: 1
    Maximal time step number (coarse grids): 0
    Minimum residual (coarse grids): 1e-07
Timestepping Start/Stop -----: -
    Output period: 50
    Maximal time step number: 10000
#    Minimum residual: 1e-7
    Matching period: 150
Timestepsize -----: -
    CFL number: 1 #3 #5 #6
#    CFL number (coarse grids): 2.2 #3
#    CFL number (large grad p): 2
    Infinite timestep (0/1): 0
    Project time steps (0/1): 1
    #Time step smoothing factor: 0
Smoother -----: -
    Residual smoother: Point_explicit
    Correction smoother: Point_explicit
    Correction smooth epsilon: 0.2
    Residual smooth epsilon: 0.2002
    Correction smoothing steps: 2
    Residual smoothing steps: 2
    Smoothing relaxation steps: 2
    No smoothing near shocks (0/1): 0
Perfect gas thermodynamic -----: -
    Gas constant R: 287
    Gas constant gamma: 1.4
Transport coefficients -----: -
    Prandtl number: 0.72
    Sutherland constant: 110.4
    Sutherland reference viscosity: 1.716e-05
    Sutherland reference temperature: 273
Geometry -----: -
    Grid scale: 1.0
    Reference relation area: 0

```

```

Reference length (pitching momentum): 1.0
Reference length (rolling/yawing momentum): 1
  Origin coordinate x: 0
  Origin coordinate y: 0
  Origin coordinate z: 0
References -----: -
  Reference temperature: 293.15
  Reynolds length: 0.0127
Reference velocity: 44.2
  Reference pressure: 101325
Preconditioning -----: -
  Preconditioning: PrimNew
  Cut-off value: 1.0 #2.0
Turbulence -----: -
  Turbulence model version: k-w #RSM

k-w models -----: -
  k-w model version: Menter_SST.1994

Dual time -----: -
  Unsteady time stepping: dual
  Unsteady activate inner iteration output (0/1): 0
  Unsteady show pseudo time steps (0/1): 1
  Unsteady physical time step size: 1.5e-4
  Unsteady physical time offset: 0
  Unsteady computational time step size: -1
  Unsteady physical time steps: 480
  Unsteady inner iterations per time step: 500
  Minimum number of inner iterations per time step: 100
  Unsteady implicit scheme order: 2
  Unsteady extrapolation order: 0 #1
  Compute harmonics on surface (0/1): 0
Flow time averaging -----: -
  Compute flow statistics: mean+meanvelgrad+variance
  Reinitialize flow averaging: 1 1 1

Cauchy convergence control -----: -
  Use Cauchy convergence control: relative
  Error for Cauchy convergence clift: 1e-4 #-1
  Error for Cauchy convergence cdrag: 1e-5 #-1
  Error for Cauchy convergence cmy: -1
  Error for Cauchy convergence cost function: 1e-06
  Number of samples for Cauchy convergence: 40
  Factor for dynamic Cauchy convergence: 0.05
Moving grid -----: -
  Type of grid movement: static
Motion description filename: (none)
Motion hierarchy filename: (none)
Extended motion monitoring (0/1): 0
Geometric conservation law (0/1): 1
  Initialize deformation (0/1): 0
Flutter parameters description filename: (none)
Monitoring -----: -
  Monitor history (0/1): 1
  Residual monitoring type (0/1): 1

```

```
#           Monitoring values: Residual_dr11/dt_dr22/dt_dr33/dt_dr13/dt_deps-rsm/dt_C-lift_
C-drag_C-my_Max-res_Max-y+_Max-eddyv
           Monitoring values: Residual_drk/dt_drk2/dt_Max-res_C-lift_C-drag_C-my_Max-y+_
Max-eddyv
#           Monitoring values: Residual_dnuv/dt_Max-res_C-lift_C-drag_C-my_Max-y+_Ma
x-eddyv
           #_C-lift-p_C-lift-v_C-drag-p_C-drag-v
           Global monitoring (0/1/2): 1 #2
           Monitoring significant figures: 4_4_4_4_4_4_4_4_4
           Extended coefficient monitoring (0/1): 0
Logfile control -----: -
           Enable logfile output on all domains (0/1): 0 #1
```

tau2plt

```
Volume element options -----: -
           Volume data output (0/1): 1
           Element types for zone: (none)
           One zone for all volume elements: 0
           Percentage of BL reduction (0-100): 20
           Specify tetrahedral list: (none)
           Specify pyramide list: (none)
           Specify prism list: (none)
           Specify hexaeder list: (none)
Surface element options -----: -
           Surface data output (0/1): 1
           Create surface zone for surface element: (none)
           Create surface zone for boundary marker: (none)
           Create one boundary: 0
           Create surface zone for triangular elements: (none)
           Create surface zone for quadrilateral elements: (none)
Output Control -----: -
           Output format: tecplot
           Ascii (0/1): 0
           Precision : 9
           Title of output file: (none)
Other -----: -
           Variable list: (none)
           Swap yz: 0
           Separate original and adapted file: 0
           Old grid size: 0
           Use node ID: 0
           Output level: 5
```

Updates
