



# ESCUELA POLITÉCNICA NACIONAL

## FACULTAD DE CIENCIAS

### METHODS OF TIME SERIES DATA MINING

### CORRELACIONES ENTRE ESPECTROS DE SERIES TEMPORALES MAGNÉTICAS DEL MÉTODO EMPIRICAL MODE DECOMPOSITION PARA DATOS DE LA TIERRA Y EL SOL

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE FÍSICO

**LUIS FERNANDO PALACIOS FLORES**

[luis.palacios01@epn.edu.ec](mailto:luis.palacios01@epn.edu.ec)

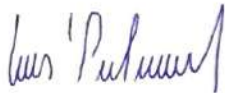
**DIRECTOR: RAMÓN XULVI BRUNET**

[ramon.xulvi@epn.edu.ec](mailto:ramon.xulvi@epn.edu.ec)

**DMQ, SEPTIEMBRE 2022**

## CERTIFICACIONES

Yo, LUIS FERNANDO PALACIOS FLORES, declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



---

Luis Fernando Palacios Flores

Certifico que el presente trabajo de integración curricular fue desarrollado por Luis Fernando Palacios Flores, bajo mi supervisión.



---

Ramón Xulvi Brunet  
**DIRECTOR**

## **DECLARACIÓN DE AUTORÍA**

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el(los) producto(s) resultante(s) del mismo, es(son) público(s) y estará(n) a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Luis Fernando Palacios Flores

Ramón Xulvi Brunet

## DEDICATORIA

A toda alma insatisfecha que no detiene su movimiento.

“

...

*If you can meet with Triumph and Disaster  
And treat those two impostors just the same;*

...

*Yours is the Earth and everything that's in it,  
And—which is more—you'll be a Man, my son!”*

- Rudyard Kipling, *If*

## **AGRADECIMIENTO**

A toda mi familiar, por todo su cariño y por creer en mí. En especial a mi padre, porque sin todo su apoyo, yo ya no estaría aquí.

A los amigos que pude hacer en este breve momento de mi vida, entre ellos: Georgie, Daniel, André, Jorge, Cristopher, Valeria, Vladimir, Camila. Sin ellos la universidad no hubiera sido lo mismo.

Al mi tutor, el Dr. Ramón Xulvi, por todo el tiempo dedicado a la realización de este trabajo. Agradezco también por aportar más que solo conocimiento sino también por darme la oportunidad de pensar de manera diferente y continuar con mi crecimiento tanto académico como personal.

Al Dr. Luca Sorriso-Valvo por su colaboración en el desarrollo de este trabajo.

En sí, a todos los que a lo largo de mi periodo en la universidad me acompañaron, apoyaron y me permitieron crecer tanto en lo académico como en lo personal.

---

# Índice general

---

<b>ÍNDICE GENERAL</b> . . . . .	VI
<b>ÍNDICE DE FIGURAS</b> . . . . .	VIII
<b>RESUMEN</b> . . . . .	IX
<b>ABSTRACT</b> . . . . .	X
<b>1. Descripción del componente desarrollado</b> . . . . .	<b>1</b>
1.1. Marco teórico . . . . .	1
1.1.1. Definición de un plasma . . . . .	1
1.1.2. Descripciones de un plasma . . . . .	1
1.1.3. El Sol y el Viento Solar . . . . .	2
1.1.4. Misiones de Exploración Espacial y Series Temporales . . . . .	3
1.1.5. Turbulencia . . . . .	3
1.2. Motivación . . . . .	4
1.3. Objetivos . . . . .	5
1.3.1. Objetivo General . . . . .	5
1.3.2. Objetivos Específicos . . . . .	6
1.4. Alcance . . . . .	6
<b>2. Metodología</b> . . . . .	<b>7</b>
2.1. Descripción de los datos . . . . .	7
2.2. Empirical Mode Decomposition . . . . .	8
2.2.1. Cubic Splines . . . . .	11
2.2.2. Condiciones de frontera en el cálculo de las envolventes . . . . .	13

2.2.3. Ceros de una señal . . . . .	13
2.3. Cross-Correlation . . . . .	14
2.4. Barrido sobre el lapso de las ventanas de tiempo y el número de IMFs . . . . .	15
<b>3. Resultados, conclusiones y recomendaciones</b>	<b>20</b>
3.1. Resultados . . . . .	20
3.1.1. Implementación del EMD en Fortran . . . . .	20
3.1.2. Correlaciones a baja frecuencia entre ventanas de tiempo de las series temporales . . . . .	23
3.2. Conclusiones y recomendaciones . . . . .	30
<b>A. Anexos</b>	<b>32</b>
A.1. Código en Fortran con el método EMD . . . . .	32
A.2. Código en Python para realizar el barrido sobre las ventanas de tiempo y el número de IMFs . . . . .	38
<b>Bibliografía</b>	<b>41</b>

---

## Índice de figuras

---

2.1. Datos del campo magnético en la sonda PSP durante el intervalo de tiempo 03:05 - 05:15 UT el 27 de septiembre de 2020. De arriba hacia abajo: $B_R$ y $ B $ , $B_T$ y $B_N$ . . . . .	7
2.2. Datos del campo magnético en la sonda SolO desde el 01 de octubre de 2020 a las 21:20 UT hasta el 02 de octubre de 2020 a las 01:20 UT. De arriba hacia abajo: $B_R$ y $ B $ , $B_T$ y $B_N$ . . . . .	8
2.3. Envoltura de los máximos $e_{max}(x)$ y mínimos $e_{min}(x)$ locales, y su valor medio $e_m(x)$ , de la función de prueba $f$ mediante la interpolación cubic splines. . . . .	12
2.4. Mejora en el cálculo de la envoltura de los máximos $e_{max}(x)$ y mínimos $e_{min}(x)$ locales, y su valor medio $e_m(x)$ , de la función $f$ mediante la interpolación cubic splines al considerar las condiciones de frontera de los extremos. . . . .	13
2.5. Cross-Correlation entre datos de la componente radial del campo magnético. Izquierda: correlación, en una ventana de 54 minutos, entre los intervalos de mayor correlación para los datos de PSP y SolO. Derecha: autocorrelación para los datos de PSP. . . . .	15
2.6. Esquema del desplazamiento de una ventana de tiempo sobre las series temporales y el cálculo de la función cross-correlation entre pares de ventanas de tiempo. . . . .	16
2.7. Cross-correlation entre los datos de PSP y SolO en una ventana de 27 minutos. De arriba hacia abajo: ventana en PSP, ventana en SolO, función cross-correlation. . . . .	17
2.8. Similar a 2.7 aunque con los datos (de las ventanas) reconstruidos con las primeras 5 IMFs de baja frecuencia. . . . .	18



3.1. Comparación en el número de IMFs resultantes de la descomposición de las series temporales del campo magnético mediante el método EMD en Fortran, Python y Matlab. Izquierda: PSP; derecha: SolO. . . . .	20
3.2. Resultado de la descomposición de $B_R$ de PSP. De arriba hacia abajo: serie temporal de la componente radial del campo magnético; IMFs de la descomposición del método EMD y su residuo. Izquierda: Fortran, 12 IMFs; derecha: Python, 10 IMFs. . . . .	21
3.3. Reconstrucciones de la serie temporal $B_R$ de PSP a partir de sus IMFs. De arriba hacia abajo: señal, reconstrucción de Python con 80% IMFs, reconstrucción de Fortran con 75% IMFs, valor absoluto de la diferencia entre las reconstrucciones. . . . .	22
3.4. Similar a la figura 3.3 pero para los datos de SolO. . . . .	22
3.5. Similar a la figura 3.2 pero para los datos de SolO. Izquierda: Fortran, 11 IMFs; derecha: Python, 10 IMFs. . . . .	23
3.6. Máximos de correlación en el barrido sobre la duración de las ventanas de tiempo y el número de IMFs. Entre líneas puntadas se separan tres rangos de ventanas de tiempo para los que se observaron diferentes patrones de correlación relevantes en las series temporales de $B_R$ . . . . .	25
3.7. Mapas de correlaciones para algunas ventanas de tiempo en las series temporales de $B_R$ de PSP y SolO. Las correlaciones se calcularon entre datos reconstruidos con 4 IMFs de baja frecuencia. De arriba hacia abajo y de izquierda a derecha: ventanas de tiempo de 20, 28, 36, 44, 52 y 80 minutos. . . . .	26
3.8. Máximos de correlación respecto del número de IMFs de menor frecuencia que reconstruyen las ventanas de tiempo de las series temporales de $B_R$ . Las ventanas de 44 y 54 minutos se indican con marcadores diferentes como ejemplo de la tendencia. . . . .	27
3.9. Similar a 3.8 pero para las particiones de mayor correlación que se obtienen con la serie temporal completa, es decir, las particiones son fijas en el barrido. . . . .	28
3.10. Reconstrucción de las series temporales de $B_R$ con las 4 IMFs de menor frecuencia en una ventana de 54 minutos. Arriba: datos de PSP en el intervalo de tiempo 03:15-04:09 UT el 27 de septiembre de 2020; abajo: datos de SolO en el intervalo de tiempo 22:20-23:14 UT el 01 de octubre de 2020. . . . .	29
3.11. Similar a 3.10. Arriba: datos de PSP en el intervalo de tiempo 04:01-04:55 UT el 27 de septiembre de 2020; abajo: datos de SolO en el intervalo de tiempo 21:56-22:50 UT el 01 de octubre de 2020. . . . .	29

## RESUMEN

El viento solar es un plasma turbulento que lleva el campo magnético del Sol a medida que se expande a través del espacio interplanetario. En este trabajo se estudian las correlaciones entre las mediciones de las sondas espaciales Parker Solar Probe y Solar Orbiter, en el seguimiento de una parcela de plasma a medida que evoluciona radialmente desde el Sol hasta los alrededores de la Tierra. El método Empirical Mode Decomposition se usa para filtrar las altas frecuencias y reducir la influencia de la turbulencia. Los resultados sugieren que las series temporales pueden ser reconstruidas con la mitad de las componentes oscilatorias de menor frecuencia y conservar su comportamiento, realizando un buen filtrado de las altas frecuencias y obteniendo valores significativos en el cálculo de su correlación. Además, se observó la existencia de pares de intervalos de tiempo, en las series temporales, cuya correlación es alta, y que están relacionados con los datos del mismo plasma.

***Palabras clave:*** Viento solar, turbulencia, altas frecuencias, función cross-correlation, método Empirical Mode Decomposition.

## ABSTRACT

The solar wind is a turbulent plasma that carries the Sun's magnetic field as it expands through interplanetary space. Here, correlations between the measurements of the Parker Solar Probe and Solar Orbiter spacecrafts are studied, in the tracking of a plasma parcel as it evolves radially from the Sun to the surroundings of the Earth. The Empirical Mode Decomposition method is used to filter out high frequencies and reduce the influence of turbulence. Results suggest that time series can be reconstructed with half of the lower frequency oscillatory components and preserve their behavior, performing a good filtering of high frequencies and obtaining significant values in the calculation of their correlation. Furthermore, it was observed the existence of pairs of time intervals, in the time series, whose correlation is high, and which are related to data of the same plasma.

***Keywords:*** Solar wind, turbulence, high frequencies, cross-correlation function, Empirical Mode Decomposition method.

# Capítulo 1

---

## Descripción del componente desarrollado

---

### 1.1. Marco teórico

#### 1.1.1. Definición de un plasma

Un plasma consiste en una colección de partículas cargadas (iones y electrones), además de átomos y moléculas neutras en ciertos casos [1-3]. La formación de un plasma se produce cuando aumenta la temperatura de un gas neutro, o cuando se somete a campos eléctricos muy grandes, hasta que sus átomos y moléculas se ionizan [1, 2]. Por lo anterior, un plasma también puede considerarse como un gas ionizado. Por otro lado, la naturaleza cargada de las partículas de un plasma hace que sus propiedades sean diferentes a un gas normal, un líquido o un sólido, razón por la que se conoce como el cuarto estado de la materia [1]. Es más, la mayoría de la materia ordinaria del universo se encuentra en estado de plasma [2].

#### 1.1.2. Descripciones de un plasma

En física de plasmas existen varios modelos para describir a un plasma. Dependiendo de las condiciones supuestas y de las diferentes aproximaciones que se realicen, un plasma se puede estudiar a través del movimiento de partículas individuales sujetas a la fuerza de Lorentz, mediante funciones de distribución a partir de la ecuación de Boltzmann debido al gran número de partículas presentes en el plasma, o por medio de las ecuaciones de la mecánica de fluidos, en conjunto con las ecuaciones del electromagnetismo, enfocándose en las propiedades colectivas del plasma [1, 2, 4].

El enfoque fluido es muy importante al proporcionar una descripción *macroscópica* de un plasma. Esta descripción es útil por las propiedades comunes entre un plasma y

un fluido conductor, mostrando movimientos colectivos como, por ejemplo, ondas [4]. En este modelo se considera a todo el plasma como un fluido de un solo tipo, enfocándose en las propiedades colectivas y despreciando la diferencia entre las partículas cargadas [1, 2]. Este enfoque se conoce como magnetohidrodinámica y es muy útil para describir un amplio rango de comportamientos de un plasma, abarcando la física espacial y de fusión [1].

### 1.1.3. El Sol y el Viento Solar

El Sol es la principal fuente de energía que mantiene la vida en el planeta Tierra. Es un plasma conformado con alrededor de 90 % de hidrógeno, 10 % de helio y una pequeña proporción de otros elementos tales como carbono, nitrógeno, y oxígeno [3]. La temperatura en su interior ronda los  $15 \times 10^6 K$ , permitiéndole producir energía a través de la fusión nuclear de hidrógeno en helio [5]. Una pequeña proporción de la energía generada viaja alrededor de  $1.496 \times 10^8 km$  para llegar desde el Sol hasta la Tierra [6]. Esta distancia define la unidad astronómica (UA).

La atmósfera del Sol se encuentra dividida en la fotosfera, la cromosfera, y la corona [3, 5]. Esta última capa se sitúa sobre la otras dos y se extiende hacia el espacio interplanetario [6]. La expansión de la corona produce un flujo de partículas cargadas que son expulsadas por el Sol, comúnmente conocido como viento solar, y se descubrió en los comienzos de la era espacial [3]. El viento solar es una consecuencia natural de estrellas calientes como el Sol, pues cuando la temperatura de la corona es de alrededor de  $10^6 K$ , casi el 50 % de los electrones tienen velocidades térmicas que exceden la velocidad de escape gravitacional del Sol, aunque en el caso de los protones es de menos del 1 % [3, 7], debido a su mayor masa. Sin embargo, el exceso de carga positiva se contrarresta por la generación de un enorme campo eléctrico que provoca la aceleración de protones, expulsándolos del Sol [7]. Por tanto, la expansión de la corona es una manera en la que el Sol mantiene su neutralidad de carga [3].

El Sol tiene un campo magnético que a grandes escalas se aproxima a un dipolo magnético [5], cuya intensidad superficial promedio es de alrededor de  $10^{-4} T$  [3, 5, 6]. En su expansión a través del espacio interplanetario, el viento solar lleva consigo el campo magnético del Sol que se extiende radialmente hacia afuera [6] y cubre todo el sistema solar, protegiendo a los planetas de las partículas de alta energía del espacio interestelar. El límite en donde se encuentra el viento solar con el viento interestelar se llama heliopausa, mientras que la región en su interior se denomina heliosfera [3, 7], debido a la mayor influencia del Sol. Mediciones realizadas por magnetómetros en sondas espaciales muestran que la intensidad de campo magnético interplanetario alrededor de la Tierra es de unos pocos  $10^{-9} T$  [3, 6].

#### 1.1.4. Misiones de Exploración Espacial y Series Temporales

Las sondas espaciales permiten mejorar la comprensión del espacio a través de la recopilación de información de gran interés científico. Los instrumentos a bordo son capaces de realizar mediciones y observaciones que en la Tierra no serían posibles. En particular, las misiones Parker Solar Probe (PSP), a cargo de la NASA y lanzada en 2018 [8], y Solar Orbiter (SolO), dirigida por la ESA y lanzada en 2020 [9], están encargadas de estudiar la heliosfera interior, enfocándose en el Sol. Estas sondas espaciales llevan instrumentos de detección a distancia (por ejemplo, telescopios) e in-situ (por ejemplo, magnetómetros), para realizar observaciones de la corona solar y para medir el viento solar, las partículas energéticas y los campos electromagnéticos [8, 9], respectivamente. En conjunto, las sondas PSP y SolO posibilitan profundizar el entendimiento sobre la atmósfera del Sol y su actividad [10].

Los instrumentos de las sondas espaciales registran los valores de diferentes magnitudes físicas a intervalos de tiempo iguales. A este tipo de mediciones se les conoce como series temporales [11, 12]. De las series temporales registradas por las sondas espaciales son de especial interés las mediciones del campo magnético.

La dinámica de los plasmas espaciales es un proceso intrínsecamente no lineal [6, 13, 14] y en general, no estacionario [11, 12, 15]. Esto último significa que las propiedades estadísticas (por ejemplo, la media y la varianza) de las series temporales cambian con el tiempo [11, 12] y, por consiguiente, los datos no son estacionarios. Por otro lado, se dice que un sistema lineal tiene “datos lineales”.

#### 1.1.5. Turbulencia

El movimiento de los fluidos se puede dividir en dos tipos, laminares o turbulentos. El primer caso implica un movimiento regular y suave del fluido, es decir, un movimiento “predecible” [16-18]. Por otro lado, un flujo turbulento se caracteriza por tener un carácter aleatorio e impredecible [16-18]. Por ejemplo, el flujo de humo de un cigarro que se deja consumir es laminar al principio, pero a medida que el humo se desplaza se vuelve turbulento formando remolinos de diversos tamaños que se mueven aleatoriamente [17].

La turbulencia es un fenómeno que se puede manifestar en cualquier fluido, pues el movimiento de un fluido es casi siempre inestable [18]. No existe una definición rigurosa sobre la turbulencia, pero se distinguen características que permiten cierto entendimiento. El carácter impredecible de la turbulencia se debe a la sensibilidad a las condiciones iniciales, lo cual refleja su naturaleza caótica [18]. De igual manera, grandes fluctuaciones en la velocidad del fluido son propias de la turbulencia [17, 18]. Además, es común la presencia de remolinos que se forman y desplazan de manera aleatoria, como se mencionó

en el ejemplo anterior. Finalmente, en un flujo turbulento, como resultado de la difusión del fluido, se produce disipación de energía, desde remolinos grandes hacia remolinos cada vez más pequeños, en la denominada cascada de energía de Richardson [13, 18].

## Turbulencia en el Viento Solar

Los plasmas espaciales y los fluidos atmosféricos son sistemas turbulentos en evolución dinámica, cuyas interacciones son no lineales [6]. La naturaleza de las interacciones en estos sistemas es evidente cuando se tiene presente que algunas de las ecuaciones de la mecánica de fluidos son ecuaciones diferenciales no lineales parciales acopladas. En el caso del viento solar, las mediciones realizadas por sondas espaciales han demostrado que la turbulencia es un fenómeno que muestra la presencia de fluctuaciones a pequeña escala (espaciales y temporales) en el campo magnético, el campo de velocidades y la densidad de protones [13, 19, 20]. En el viento solar, la turbulencia se desarrolla fuertemente durante el desplazamiento y expansión a través del espacio interplanetario [13, 21, 22], similar al ejemplo del humo de cigarro.

## 1.2. Motivación

El alineamiento de sondas espaciales es un evento poco común del cual solamente se tienen unos pocos casos registrados [23-28] desde que se empezaron a realizar mediciones in situ de la actividad magnética del Sol hace unas pocas décadas. Estos sucesos son de gran importancia para la investigación de las características de los plasmas espaciales porque permiten estudiar la evolución del plasma a través de la heliosfera.

En particular, el seguimiento de un plasma espacial permite estudiar las propiedades de la turbulencia del viento solar, lo cual es crucial para el entendimiento de los plasmas y en general, de los fluidos. Esto se realiza analizando la influencia de la turbulencia sobre las variables que se registran en series temporales. Recientemente, las sondas Parker Solar Probe y Solar Orbiter, tuvieron su primer alineamiento radial a una distancia de  $\sim 0.1UA$  y  $\sim 1UA$  [28], respectivamente. Las mediciones del campo magnético del viento solar mostraron que el plasma pasa de un estado poco turbulento cerca del Sol hacia un comportamiento con una turbulencia altamente desarrollada e intermitente en las cercanías de la Tierra [28]. La motivación detrás de este trabajo es investigar la evolución de la parte no turbulenta del campo magnético cuando el viento solar se desplaza radialmente desde el Sol hasta la Tierra. Esta parte no turbulenta se puede cuantificar mediante un filtrado de las altas frecuencias de las series temporales asociadas.

El tipo de análisis que se realiza sobre las series temporales es crucial en cuanto a las

características que se pretenden estudiar. La correlación es una medida estadística de uso frecuente para analizar series temporales. Esta mide el grado de similitud entre dos series temporales [12]. Por ejemplo, en el seguimiento de un plasma espacial, las series temporales de cada sonda deben estar altamente correlacionadas para concluir que pertenecen al mismo plasma.

Es común que el análisis de las series temporales de los plasmas espaciales se realice mediante métodos estándar como aquellos basados en la transformada de Fourier o transformadas wavelet [12]. Sin embargo, los primeros solamente son útiles para datos estacionarios [12], mientras que los últimos están restringidos a datos con escalas temporales definidas [29, 30]. Además, algunos de los métodos estándar son estrictamente útiles para datos que provienen de sistemas lineales, o son muy especializados para explorar nuevas características en los datos. Por otro lado, las series temporales de las magnitudes físicas del viento solar son no lineales, no estacionarias y dependen de la escala [15, 31-34], es decir, de los intervalos espaciales y temporales en donde ocurren ciertos procesos físicos.

El método Empirical Mode Decomposition (EMD) puede ser útil para el análisis de series temporales magnéticas [15, 31-34]. Este método descompone una serie temporal en una suma de componentes oscilatorias denominadas funciones de modo intrínseco (IMFs), que permiten una buena reconstrucción de la serie. Reconstrucciones parciales (no todas las IMFs) hacen posible filtrar las componentes de alta frecuencia de las series temporales. Debido al carácter empírico del método, las características locales de la escala de tiempo de los datos son consideradas [29], y esto hace posible estudiar un mayor tipo de datos. Por esta razón, en este trabajo se utiliza el EMD para analizar las series temporales del viento solar durante el alineamiento de las sondas espaciales PSP y SolO.

## 1.3. Objetivos

### 1.3.1. Objetivo General

En este estudio se busca identificar posibles correlaciones relevantes entre las reconstrucciones a baja frecuencia de las series temporales de campo magnético del viento solar, en los alrededores del Sol y la Tierra (medidas por PSP y SolO, respectivamente), obtenidas mediante las IMFs de la descomposición del método Empirical Mode Decomposition. De esta manera se quiere estimar si se puede disminuir la aparente aleatoriedad que aporta la turbulencia sobre las series temporales y conservar resultados significativos.



### 1.3.2. Objetivos Específicos

- A. Implementar y aplicar el método Empirical Mode Decomposition a las series temporales.
- B. Implementar el método Cross-Correlation y aplicarlo entre reconstrucciones de las series temporales realizadas a partir de sus IMFs de menor frecuencia.

## 1.4. Alcance

El presente trabajo pretende ser un estudio exploratorio de la evolución de la parte no turbulenta de las series temporales del campo magnético al filtrar las componentes de alta frecuencia mediante el EMD. Además, se indaga en el número de IMFs necesario para reconstruir las series temporales y obtener resultados significativos en el cálculo de las correlaciones entre las series.

En este trabajo se utilizaron datos que derivan de otro estudio de series temporales de campo magnético, que también corresponden con el alineamiento de las sondas espaciales PSP y SolO [28]. Los datos recibidos fueron preprocesados por el grupo de investigación anterior por lo que se pudo omitir este paso en el análisis de las series temporales. Esto permitió continuar con la implementación y aplicación del método EMD, que en este caso se realizó en Fortran. También se utilizaron las librerías de Matlab y Python para comparar y validar la implementación realizada.

El método Cross-Correlation es uno de los métodos estándar más utilizados para correlacionar series temporales [12]. Este método se implementó en Python debido a la eficiencia de tiempo, manejo y almacenamiento de variables, para el caso particular del cálculo de las correlaciones propuestas.

# Capítulo 2

---

## Metodología

---

### 2.1. Descripción de los datos

Las series temporales de campo magnético del viento solar provienen de mediciones in situ realizadas por las sondas espaciales PSP y SolO durante su primer alineamiento radial a una distancia de  $\sim 0.1UA$  y  $\sim 1UA$ , respectivamente. Los datos de la sonda PSP fueron registrados el 27 de septiembre de 2020 desde las 03:05 UT hasta las 05:15 UT (130 minutos), y los datos de la sonda SolO fueron tomados el 01 de octubre de 2020 a las 21:20 UT hasta el 02 de octubre de 2020 a las 01:20 UT (240 minutos). La frecuencia de muestreo de los datos estudiados es de  $8 Hz$ .

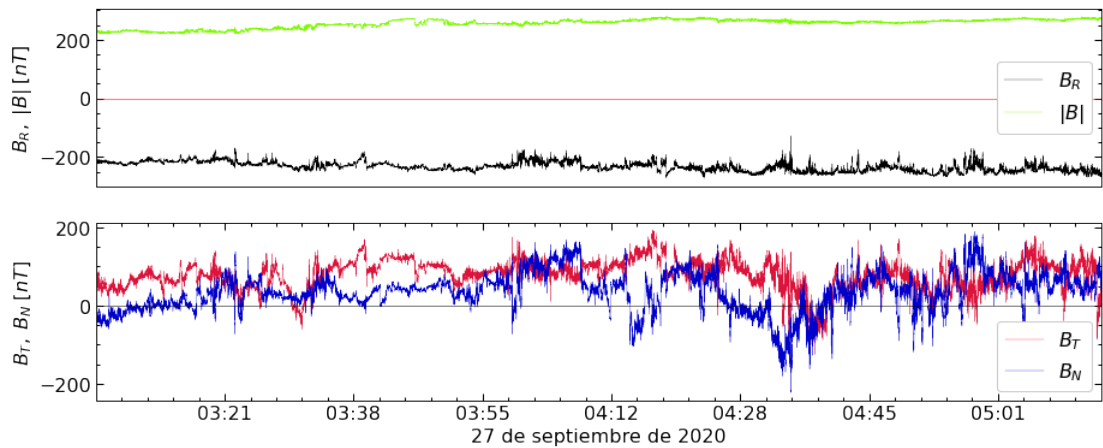


Figura 2.1: Datos del campo magnético en la sonda PSP durante el intervalo de tiempo 03:05 - 05:15 UT el 27 de septiembre de 2020. De arriba hacia abajo:  $B_R$  y  $|B|$ ,  $B_T$  y  $B_N$ .

En las figuras 2.1 y 2.2 se muestra la intensidad de campo magnético ( $|B|$ ) y sus componentes en el sistema de coordenadas RTN, para las series temporales de ambas

sondas espaciales. En el sistema de coordenadas RTN, la componente R apunta radialmente desde el centro del Sol hasta la nave espacial; la componente tangencial T es el producto cruz de la velocidad angular del Sol con R; y la componente normal N completa la triada ortogonal y tiene la misma dirección que la velocidad angular del Sol [13].

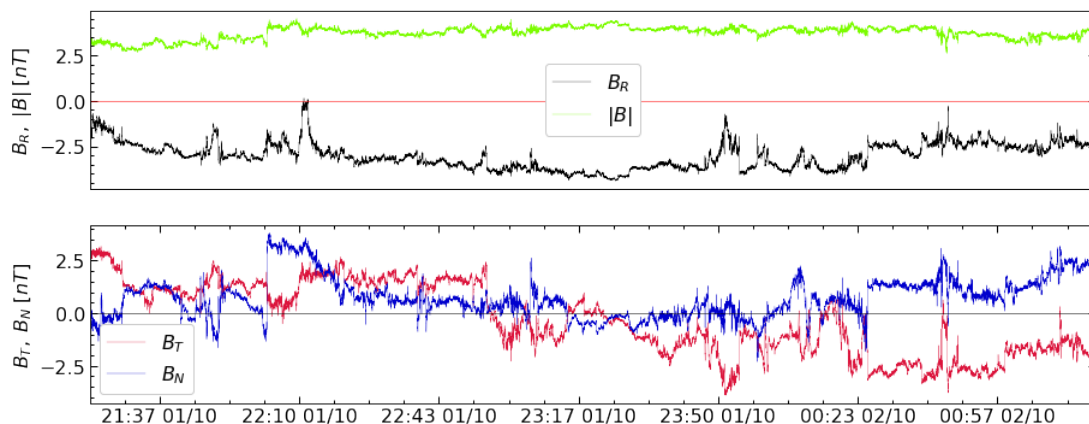


Figura 2.2: Datos del campo magnético en la sonda SolO desde el 01 de octubre de 2020 a las 21:20 UT hasta el 02 de octubre de 2020 a las 01:20 UT. De arriba hacia abajo:  $B_R$  y  $|B|$ ,  $B_T$  y  $B_N$ .

## 2.2. Empirical Mode Decomposition

Muchos de los fenómenos físicos en la naturaleza se caracterizan por tener una dinámica no lineal [35] y, en muchas ocasiones, no estacionaria. Incluso el ejemplo más sencillo que se le puede ocurrir a un físico, como es el movimiento de un péndulo simple en ausencia de fuerzas externas y rozamiento es, en general, no lineal. Además, aún en el caso de pequeñas oscilaciones, cuando se considera un movimiento amortiguado, se pierde la estacionariedad, pues la amplitud de oscilación deja de ser constante y su valor medio cambia con el tiempo [11, 12].

Los datos de series temporales pueden mostrar diversos patrones que son más fáciles de entender cuando se descomponen en varios componentes. Es como observar un reloj y tener una idea de cómo funciona, pero solamente entenderlo (o mejorar la comprensión) cuando se lo desarma y observa la posición de sus partes y cómo estas interactúan entre sí. De igual manera, descomponer una señal puede hacer más fácil su análisis cuando se hace uso de las técnicas adecuadas. Por ejemplo, la transformada de Fourier permite identificar las componentes de frecuencia de una señal y, operando sobre estas, filtrar las frecuencias [36] que no sean de interés.

Los métodos estándar basados en la transformada de Fourier y la transformada Wavelet son muy utilizados para analizar series temporales pues están basados en una teoría mate-

mática robusta [29, 30], lo que facilita la interpretación de los resultados. Sin embargo, las teorías analíticas que soportan estos métodos requieren de propiedades de ortogonalidad y/o completitud, lo que fija a priori el número de componentes en las que se descompone una señal. Más aún, estas componentes pueden tener otras restricciones adicionales como ser funciones sinusoidales, el cual es el caso en la transformada de Fourier con los modos de oscilación armónicos.

En una serie temporal cualquiera puede que su descomposición no sea idónea en componentes sinusoidales. Tampoco puede que exista un número fijo para el cual se obtenga una clara distinción entre ellas. Este es el caso de las series temporales que provienen de sistemas físicos no lineales, no estacionarios y que pueden tener diferentes escalas características. Por esta razón, el análisis y la descripción de estos sistemas requiere el uso de diferentes métodos basados en técnicas adaptativas que permitan describir sus características inherentes a partir de una descomposición basada en los propios datos.

El método Empirical Mode Decomposition (EMD) es un método adaptativo que se basa en la escala de tiempo característica local de los datos, permitiendo descomponer señales de procesos no lineales y no estacionarios [29]. Es decir, este método analiza las características de los datos de punto a punto para realizar la descomposición sin importar si los procesos son no lineales o no estacionarios. Además, como el método está basado en los datos, la base de la descomposición se define a posteriori y no se imponen restricciones sobre estas componentes (amplitud constante y frecuencia fija en el caso de la expansión de Fourier). El EMD permite descomponer una señal en un número finito de componentes oscilatorias denominadas funciones de modo intrínseco (Intrinsic Mode Functions, IMFs), las cuales poseen amplitud y frecuencia moduladas.

La descomposición de una serie temporal en sus IMFs se realiza mediante un proceso iterativo denominado *sifting* (tamizado). El razonamiento detrás de este proceso consiste en extraer los componentes de una señal comenzando con aquellas de mayor frecuencia y con cada iteración separar los diferentes modos de una señal según sus características para de esta manera terminar con la IMF con menor frecuencia. Esto es similar a tener un arreglo de tamices diferentes. Al finalizar este proceso la serie temporal original  $x(t)$  puede escribirse como la suma de  $M$  modos oscilantes  $c_j(t)$  (IMFs) y un residuo monótono  $r(t)$ , que representa la tendencia a largo plazo de la señal original, lo cual es usual en el análisis de series temporales [12].

$$x(t) = \sum_{j=1}^M c_j(t) + r(t) \quad (2.1)$$

Las IMFs que se obtienen tras la descomposición poseen dos propiedades importantes.

- (1) El número de extremos y cruces por cero es igual o difieren como máximo en uno.
- (2) El valor medio es cero.

La primera condición garantiza que no existan oscilaciones complicadas en las componentes de la descomposición y que exista un cruce por cero al menos una vez antes de que pasar de un extremo a otro, lo que dota a las IMFs de simetría y hace que el método sea adaptativo. Por otro lado, la segunda condición permite obtener frecuencias instantáneas que no sean afectadas por fluctuaciones asimétricas. Esto también implica que las IMFs son estacionarias (lo cual permite ampliar el análisis de las series temporales [37]). De esta manera, cada IMF tiene una amplitud modulada y frecuencias instantáneas físicamente significativas que pueden ser de banda estrecha [29].

El proceso *sifting* es la base del método EMD y los pasos que se utilizaron en este trabajo pueden resumirse en el siguiente algoritmo:

1. Establecer la condición inicial para el residuo  $r(t) = x(t)$ .
2. Identificar los máximos y mínimos locales de  $x(t)$ .
3. Usar la interpolación cubic splines para calcular las envolventes, superior  $e_{max}(t)$  e inferior  $e_{min}(t)$ , definidas por los máximos y mínimos locales, respectivamente.
4. Calcular la envolvente media  $e_m(t) = \frac{1}{2}(e_{min}(t) + e_{max}(t))$ .
5. Evaluar  $h_i(t) = x(t) - e_m(t)$ , denominada proto-IMF, y verificar si cumple con las condiciones (1) y (2) para ser una IMF.
6. Si  $h_i(t)$  es una IMF entonces  $c_j(t) = h_i(t)$  se guarda y se actualiza el residuo  $r(t) = r(t) - c_j(t)$ , de lo contrario se repiten los pasos del 2 al 5 sobre  $x(t)' = h_i(t)$  hasta obtener una IMF.

Las iteraciones en el proceso *sifting* son limitadas puesto que de lo contrario las IMFs pierden sentido físico, dado que esto puede eliminar las características locales de las series. Desde la presentación del método EMD se han propuesto y utilizado diferentes criterios de detenimiento [31, 38, 39]. En este trabajo solamente se utiliza el criterio que fue propuesto originalmente. Este criterio establece que el proceso *sifting* se detiene cuando  $SD < SD^*$ , donde

$$SD = \sum_{t=1}^T \frac{|h_i(t) - h_{i-1}(t)|^2}{h_{i-1}^2(t)}, \quad i > 1 \quad (2.2)$$

$T$  es la duración de la serie temporal (“longitud” de los datos) y  $SD^*$  es un valor límite que puede tomar valores entre 0.2 y 0.3 [29].

Cada vez que el proceso *sifting* encuentra una IMF se verifica el número de extremos del residuo  $r(t)$ , definido en el paso 6 del algoritmo, para determinar si la descomposición finaliza. El residuo representa la tendencia media de la serie temporal  $x(t)$  y puede ser una función constante o monótona. Por tanto, el método EMD termina cuando el residuo tiene como máximo un extremo o se cumple algún otro criterio de detenimiento.

La descomposición que realiza el método EMD genera  $\sim \log_2(N)$  IMFs, donde  $N$  es el número de puntos de la serie temporal. Esto es porque en el caso de ruido estocástico de banda ancha, el EMD actúa como un banco de filtros diádicos [39] (arreglo de filtros pasa banda que reduce en un factor de 2 la salida de cada filtro [40]). A través de esta aproximación, en este trabajo se impuso un máximo al número esperado de IMFs y con la condición de que el residuo tenga a lo mucho un extremo, finalizar la descomposición de los datos.

La implementación del método EMD en Fortran se hizo en doble precisión y se adjunta en el anexo A.1. Sin embargo, en la verificación de las condiciones del residuo se redujo la precisión a la mitad debido a que de lo contrario el algoritmo seguía iterando y generando más componentes que ya no eran IMFs.

En este trabajo se tomó  $SD^* = 0.3$  como el valor límite para el criterio de detenimiento del proceso *sifting*. En la implementación realizada en Fortran, junto a esta condición, también se fijó un número máximo de iteraciones, pues de lo contrario este número se hacía muy grande. Como se mencionó, esto puede hacer que las IMFs pierdan relevancia física. Además, como los resultados no variaron mucho para un número de iteraciones similar al número esperado de IMFs ( $\sim 16$  para PSP y  $\sim 17$  para Solo), se tomaron 20 iteraciones como el número máximo de iteraciones del proceso *sifting* para descomponer los datos de las series temporales del campo magnético.

### 2.2.1. Cubic Splines

Evaluar las envolventes de los extremos locales es un paso importante para el cálculo de las IMFs. Se han estudiado algunas formas de realizar la interpolación sobre los extremos para estimar las envolventes, pero la interpolación mediante cubic splines es la más utilizada [29, 38]. A continuación, se explica brevemente la idea detrás de este método de interpolación.

La interpolación cubic splines permite ajustar localmente un conjunto de puntos a una función a trozos de tal manera que cada parte se conecte suavemente. Esta interpolación se realiza mediante polinomios de tercer grado haciendo coincidir la función y sus derivadas en los puntos de datos que conectan los trozos de la función [41]. Así, para un conjunto discreto de  $n + 1$  puntos  $(x_i, y_i)$ , descritos por una función  $f$  en el intervalo  $[a, b]$  tal que  $y_i = f(x_i)$  y  $a = x_0 < x_1 < \dots < x_n = b$ , una función  $S$  es un cubic splines si satisface las

siguientes condiciones [42]:

1. En el intervalo  $[x_i, x_{i+1}]$ ,  $S(x)$  es un polinomio de tercer grado denotado por  $S_i(x)$ ,  $i = 0, 1, \dots, n - 1$ .
2.  $S_i(x_i) = y_i$  y  $S_i(x_{i+1}) = y_{i+1}$ ,  $i = 0, 1, \dots, n - 1$ .
3.  $S'_i(x_{i+1}) = S'_{i+1}(x_{i+1})$ ,  $i = 0, 1, \dots, n - 2$ .
4.  $S''_i(x_{i+1}) = S''_{i+1}(x_{i+1})$ ,  $i = 0, 1, \dots, n - 2$ .

En cada intervalo la función de cubic splines toma la forma  $S_i(x) = a_i + b_i x + c_i x^2 + d_i x^3$ ,  $i = 0, 1, \dots, n - 1$ , y los coeficientes se determinan mediante las condiciones de frontera. En total se tienen  $4n$  incógnitas, pero  $4n - 2$  ecuaciones, por lo que para cerrar el sistema se pueden utilizar algunas condiciones adicionales. En este trabajo se utilizó las denominadas condiciones de frontera naturales, las cuales son  $S''_0(x_0) = S''_{n-1}(x_n) = 0$  [41, 42].

En la figura 2.3 se muestra el uso de la interpolación cubic splines para evaluar las envolturas de los máximos y mínimos locales de una función de prueba  $f$ . El cálculo de los extremos no considera los puntos al inicio y al final de la función, por lo que las envolturas divergen en estos puntos. Por este motivo, para que el proceso *sifting* se realice de manera adecuada, se deben imponer condiciones de frontera adicionales sobre los máximos y mínimos de los datos.

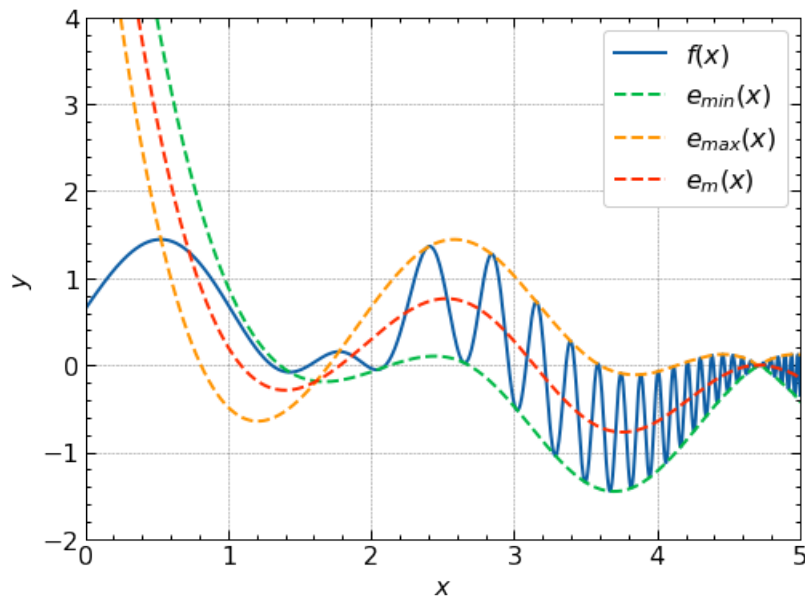


Figura 2.3: Envoltura de los máximos  $e_{max}(x)$  y mínimos  $e_{min}(x)$  locales, y su valor medio  $e_m(x)$ , de la función de prueba  $f$  mediante la interpolación cubic splines.

### 2.2.2. Condiciones de frontera en el cálculo de las envolventes

La evaluación adecuada de las envolventes al inicio y al final de los datos se resaltó desde que se presentó el método. Las condiciones de frontera seleccionadas permiten mitigar la propagación de ondas complejas desde los puntos en los extremos hacia el interior de las IMFs. Asimismo, esto es importante para que la interpolación permita obtener mejores resultados, puesto que los métodos de interpolación suelen realizar suposiciones para tener un sistema de ecuaciones cerrado. Por ejemplo, en el caso del cubic splines se suele asumir las condiciones de frontera naturales.

En este trabajo se decidió duplicar los valores de los máximos y mínimos locales en los alrededores del inicio y final de los datos hacia los bordes de la señal. Este es uno de los posibles enfoques que se han probado para el cálculo de las envolventes y ha demostrado buenos resultados [38]. En la figura 2.4 se observa que las envolventes de los máximos y mínimos locales se ajustan mejor a la función de prueba que cuando no se consideró las condiciones de frontera.

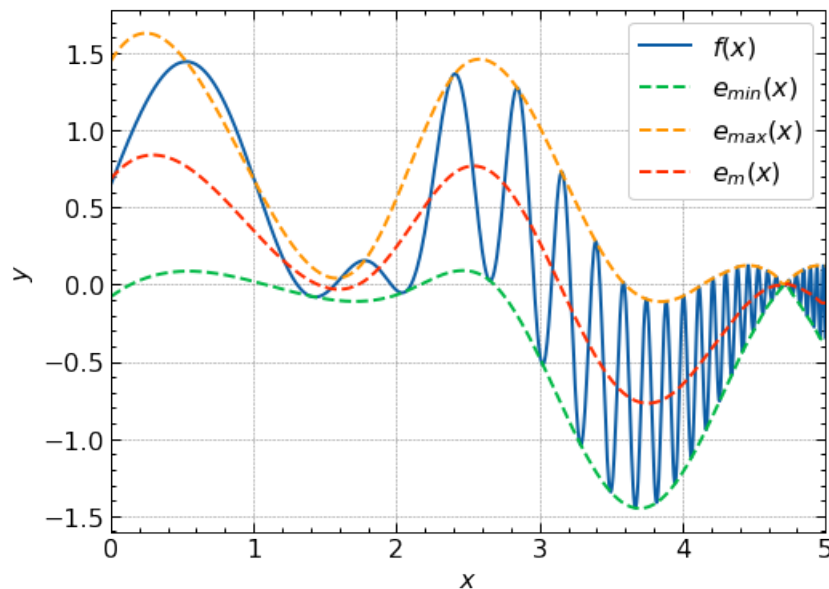


Figura 2.4: Mejora en el cálculo de la envoltura de los máximos  $e_{max}(x)$  y mínimos  $e_{min}(x)$  locales, y su valor medio  $e_m(x)$ , de la función  $f$  mediante la interpolación cubic splines al considerar las condiciones de frontera de los extremos.

### 2.2.3. Ceros de una señal

El cálculo del número de cruces por cero de una señal es necesario para verificar si el proceso *sifting* ha obtenido una IMF o no. A diferencia de estimar los valores en donde una función se hace cero, el número de veces que una señal cruza por cero es sencillo de calcular. Para esto es suficiente con determinar el signo del producto de dos puntos



consecutivos de la señal y contar el número de veces en los que este producto es negativo. Esto es debido a que el producto de dos puntos consecutivos de una función siempre es positivo cuando la función está por debajo o por encima de cero, pero cuando los puntos están en diferentes extremos del cero el producto es negativo.

### 2.3. Cross-Correlation

Las medidas de correlación permiten comparar objetivamente cuánto se parecen dos series temporales. La correlación es importante para determinar cómo se producen los cambios entre dos variables estadísticas. Si estas variables están altamente correlacionadas entonces se espera que, si una de esas variables se comporta de una manera particular, también lo hará la otra. La correlación puede ser negativa, lo que significa que cuando una de las variables incrementa, la otra disminuye [12]. Esta relación permite identificar si dos variables estadísticas son relevantes al mismo tiempo en un análisis o solamente una de ellas.

La correlación cruzada (cross-correlation) calcula la correlación entre dos series temporales por medio del desplazamiento de una serie respecto de la otra. Para un conjunto discreto de  $N$  valores de dos series temporales  $X$  y  $Y$ , la función cross-correlation entre  $X$  y  $Y$  está dada por [12]:

$$\gamma_{XY}(k) = \frac{\sum_{t=1}^{N-k} (x_t - \bar{x})(y_{t+k} - \bar{y})}{\sqrt{\sum_{t=1}^N (x_t - \bar{x})^2} \sqrt{\sum_{t=1}^N (y_t - \bar{y})^2}} \quad (2.3)$$

donde  $k$  es el desfase (lag) temporal entre los valores de las series temporales,  $\bar{x}$  y  $\bar{y}$  son los valores medios de las dos series temporales  $X$  y  $Y$ , respectivamente.

La función cross-correlation dada por la ecuación (2.3) está normalizada y toma valores entre  $-1$  y  $1$ . Cuando  $X = Y$ ,  $\gamma_{XX}(k)$  se denomina función de autocorrelación y es máxima para  $k = 0$ , es decir,  $\gamma_{XX}(0) = 1$ .

La función de correlación se evalúa para series temporales de la misma longitud. Estos cálculos pueden incluir partes de las series temporales cuya correlación es baja. Por esta razón es común calcular la función de correlación entre partes de las series temporales, estas partes tienen igual longitud. Cada una de estas partes contienen los datos correspondientes a un intervalo de tiempo y decimos que es una ventana de tiempo en los datos. De esta manera, una ventana de tiempo tiene asociada un tiempo inicial y un tiempo final, y una amplitud de tiempo o duración que la caracteriza. También, para diferenciar y ordenar las

diferentes ventanas se les puede asignar un índice. Tomar ventanas de tiempo es útil para determinar los intervalos de tiempo de mayor correlación entre dos series temporales.

En la figura 2.5 se muestran dos ejemplos del cálculo de la función cross-correlation para las series temporales de la componente radial del campo magnético  $B_R$ , sin ningún filtro de frecuencias. La curva de la izquierda muestra la función cross-correlation para el par de ventanas de 54 minutos, una de PSP y la otra de SolO, que presentaron mayor correlación. Por otro lado, en la curva de la derecha se observa el perfil característico de una función de autocorrelación, con los datos de PSP.

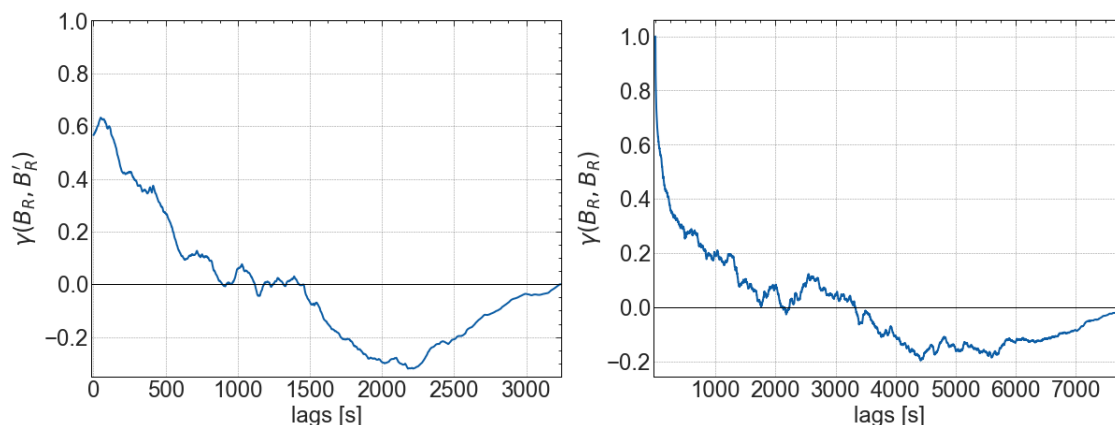


Figura 2.5: Cross-Correlation entre datos de la componente radial del campo magnético. Izquierda: correlación, en una ventana de 54 minutos, entre los intervalos de mayor correlación para los datos de PSP y SolO. Derecha: autocorrelación para los datos de PSP.

## 2.4. Barrido sobre el lapso de las ventanas de tiempo y el número de IMFs

La correlación se puede calcular con las series completas o, en el caso de filtrar algunas de sus frecuencias, con reconstrucciones parciales de las series temporales. En este trabajo, las reconstrucciones se obtienen filtrando las altas frecuencias a partir de las IMFs de cada serie. Esto permite maximizar el valor de la función cross-correlation, pues se reduce la variabilidad de las series. Por otro lado, tomar ventanas de tiempo de diferente duración también hace posible cambiar el valor de la correlación. Las correlaciones que buscamos en este trabajo utilizan ambos enfoques, es decir, se toman diferentes ventanas de tiempo y se reconstruyen las series con menos IMFs. Para entender el procedimiento primero se explica el barrido sobre las ventanas de tiempo y luego sobre las IMFs.

Supongamos que tenemos una serie temporal cuya duración total es  $T$  y una ventana de amplitud  $\Delta t$ , que empieza en  $t_s$  y termina en  $t_e = t_s + \Delta t$ . Si la ventana se desplaza sobre la serie temporal cambiando su tiempo inicial en un tiempo  $d$ , tal que  $d = t_{s+1} - t_s$ ,

entonces el número de particiones o posibles ventanas que podemos tomar de los datos está dado por:

$$n = \left\lfloor \frac{T - \Delta t}{d} \right\rfloor + 1 \quad (2.4)$$

Debido a la función suelo cualquier ventana se desplaza sobre la serie hasta una duración  $T' = \Delta t + (n - 1)d \leq T$ . También, cada ventana tiene un índice  $i$  que se corresponde con el intervalo de tiempo  $[id, id + \Delta t]$ ,  $i = 0, 1, \dots, n - 1$ . Notemos que las ventanas se pueden superponer.

Para el cálculo de la función cross-correlation se toman pares de ventanas igual duración, una de PSP y la otra de SolO. Las ventanas de las series de PSP y SolO se eligen de  $n_1$  y  $n_2$  posibles ventanas (de acuerdo a la ecuación (2.4)), respectivamente. Todos los resultados de correlación asociados a todos los posibles pares de ventanas se pueden representar en una grilla o matriz. Si las ventanas de las series temporales tienen índices  $i$  y  $j$ , entonces la componente  $ij$  de la matriz de correlaciones es la función de correlación de las series temporales en los intervalos  $[id, id + \Delta t]$  y  $[jd, jd + \Delta t]$ , donde  $i = 0, 1, \dots, n_1 - 1$ , y  $j = 0, 1, \dots, n_2 - 1$ .

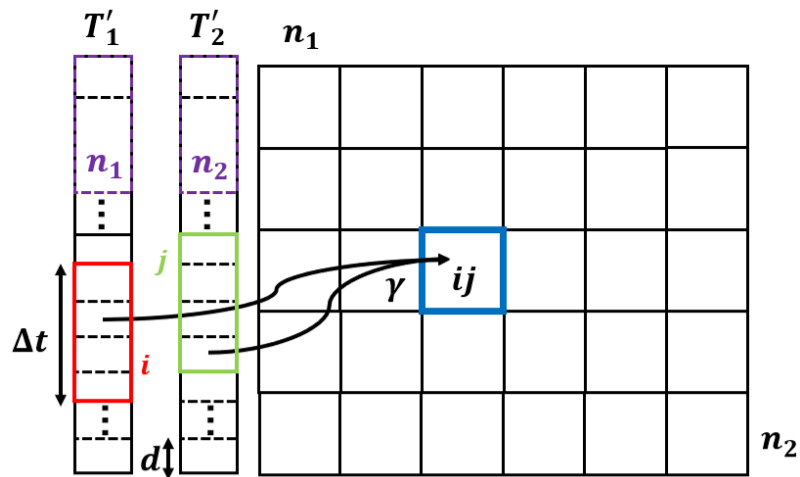


Figura 2.6: Esquema del desplazamiento de una ventana de tiempo sobre las series temporales y el cálculo de la función cross-correlation entre pares de ventanas de tiempo.

En la figura 2.6 se muestra gráficamente la idea detrás cálculo de las funciones de correlación, para una ventana de amplitud  $\Delta t$ . El tiempo de desplazamiento ( $d$ ) de cada ventana sobre las series temporales, en las  $n_1$  y  $n_2$  particiones, permite encontrar los intervalos de tiempo que están más correlacionados. La exactitud de estos intervalos está determinada por el tamaño del desplazamiento. En este trabajo se toma un valor lo suficientemente pequeño en comparación con la duración de las series temporales.

El barrido sobre el lapso de las ventanas de tiempo consiste en calcular la función cross-correlation, como se muestra en la figura 2.6, desde una ventana de amplitud mínima a una de amplitud máxima. Esto genera tantas matrices de correlaciones como amplitudes de ventanas.

Cabe señalar que los barridos propuestos se realizaron en Python, debido a la facilidad de implementación, manejo y almacenamiento de variables. En el anexo A.2 se adjunta el código en Python para realizar estas cuentas. Los resultados se almacenan en la variable `params_time` al hacer iterar la función `params_xcorr` en el ciclo que barre las diferentes ventanas de tiempo.

Ahora, veamos un ejemplo del cálculo de la función cross-correlation entre un par de ventanas. Tomando  $d = 7$  minutos y  $\Delta t = 27$  minutos entonces el número de celdas que se obtienen para PSP y SolO son  $n_1 = 15$  y  $n_2 = 31$ , respectivamente. En la figura 2.7 se muestra la función cross-correlation entre la ventana 10 de la sonda PSP y la ventana 20 de la sonda SolO. El intervalo de tiempo correspondiente en PSP es 04 : 25 – 04 : 52 UT del 27 de septiembre de 2020 y el intervalo de tiempo en SolO es 23 : 05 – 23 : 32 UT del 01 de octubre de 2020. El máximo de correlación es 0.37 y se obtuvo con un desfase de 9.67 minutos entre las series temporales.

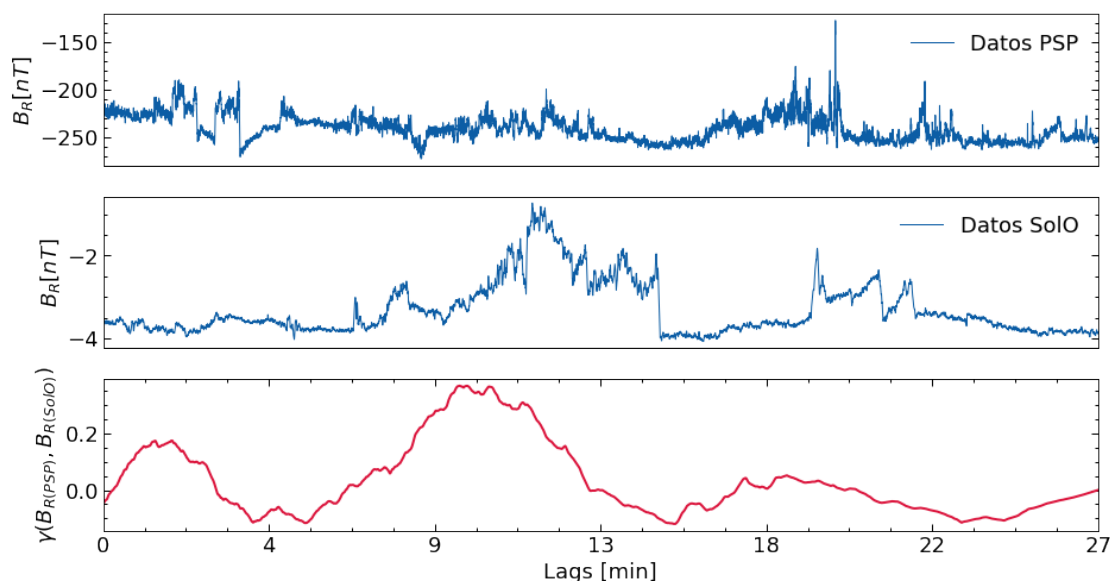


Figura 2.7: Cross-correlation entre los datos de PSP y SolO en una ventana de 27 minutos. De arriba hacia abajo: ventana en PSP, ventana en SolO, función cross-correlation.

Recordemos que el desfase es el tiempo que se desplazan los datos de una serie temporal respecto de la otra en el cálculo de la función cross-correlation. Aquí reportamos el desfase donde la función de correlación es máxima. Esta magnitud podría contener información física de relevancia, y se debería estudiar sobre aquellos pares de ventanas que presentan la máxima correlación. En nuestro estudio no se ha hecho esto, pero es algo que se podría

investigar en un análisis exploratorio futuro.

Hasta ahora hemos explicado la idea de cómo evaluamos la correlación entre ventanas de dos series. Sin embargo, uno de nuestros objetivos principales es hacer estos cálculos entre las series resultantes de un proceso de filtrado, donde las componentes de alta frecuencia hayan sido eliminadas. Así, primero nos interesa descomponer las series en sus IMFs. Luego se hacen reconstrucciones parciales, al ir disminuyendo las IMFs, hasta quedarnos con un número determinado de aquellas componentes correspondientes a las bajas frecuencias. Y tras cada reconstrucción, calculamos las correlaciones. El número de IMFs que necesitemos considerar, para obtener resultados significativos en el cálculo de las correlaciones, es una de las cosas que queremos determinar en este trabajo.

Este procedimiento se realiza para cada amplitud de ventana. También, como las series pueden descomponer en un número diferente de componentes, en este barrido, la serie con el menor número de IMFs limita las componentes filtradas de la otra serie. Cuando ambas series tienen el mismo número de IMFs entonces el filtro termina con la componente de menor frecuencia más el residuo.

En la figura 2.8 se muestran las reconstrucciones de las ventanas de tiempo de la figura 2.7 y su función cross-correlation. Cada reconstrucción se realiza con 5 IMFs. Al reducir las altas frecuencias, los valores de la función de correlación aumentan respecto de la curva de la figura 2.7. El máximo de correlación es 0.42 con un desfase de 9.70 minutos.

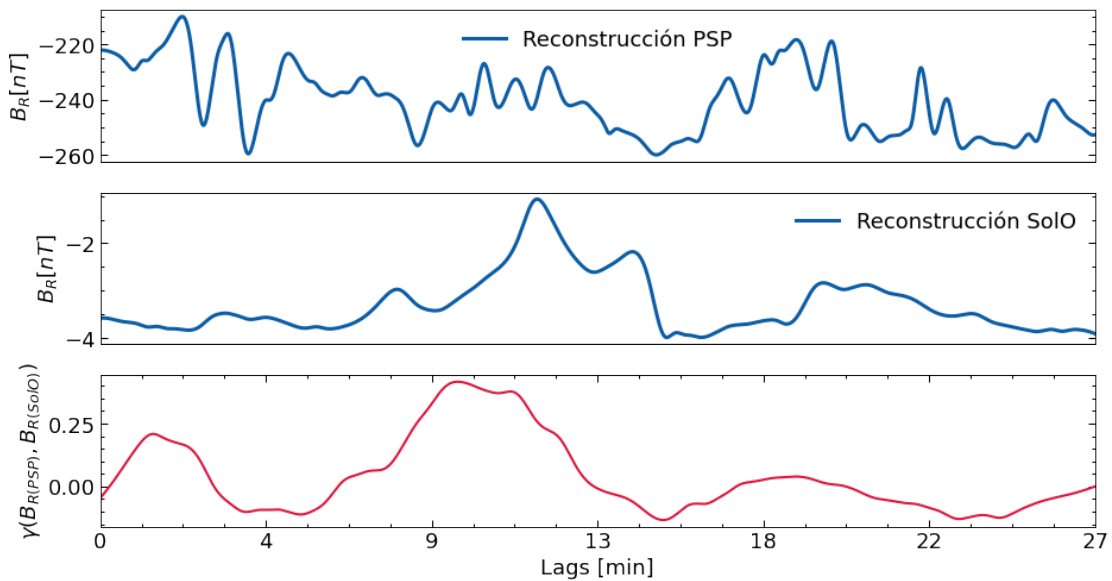


Figura 2.8: Similar a 2.7 aunque con los datos (de las ventanas) reconstruidos con las primeras 5 IMFs de baja frecuencia.

Los cálculos se ejecutaron con las IMFs obtenidas de la implementación del método EMD en Fortran y aquellas resultantes de una implementación en Python [43]. Esto se

hizo para comparar los resultados obtenidos en Fortran con los de una implementación profesional del EMD. En esta última se fijó el valor límite del criterio de detenimiento del proceso *sifting* en  $SD^* = 0.3$ , como en la implementación de Fortran.

En nuestro código de Python del anexo A.2 estas cuentas se muestran para la implementación del EMD de Python. La función `params_xcorr` ejecuta el barrido sobre las IMFs para cada valor de ventana. El residuo y las IMFs extraídas de las series se almacenan en las columnas de un arreglo bidimensional. Las columnas de este arreglo se pueden sumar fácilmente en Python, especificando las columnas a sumar. El arreglo (columna) resultante es la reconstrucción parcial de una serie.

Para cada valor de ventana, todas las posibles correlaciones entre pares de particiones de las series se realizan en un doble ciclo. En cada iteración se calcula el máximo de correlación. Estos valores se ordenan de mayor a menor a través de la función `get_jixcorrs` y se registran en la variable `max_vals`. Junto a estos valores se almacenan los índices de las particiones de cada serie. También, para cada máximo de correlación, se evalúa y se almacena el desfase entre cada par de ventanas. De esta manera, el primer valor contiene el máximo de todas las posibles correlaciones y las particiones asociadas.

Una ligera modificación de la función `params_xcorr` permite efectuar el barrido con los resultados de Fortran. En este caso, las IMFs se introducen externamente como variables globales. Primero, como ya no se utiliza la implementación del EMD en Python, se retiran los parámetros asociados. Luego, se elimina el parámetro de entrada `field_component` así como la lectura de las componentes que se realiza con este parámetro. Finalmente, en esta parte se cargan las IMFs y se limita la duración de las series temporales con los valores de  $T'$  correspondientes.

Por último, cabe señalar que las funciones mencionadas son de autoría propia y son las que permiten extraer información del barrido. Dentro de estas funciones se utilizan otras funciones propias de Python para realizar procedimientos de soporte, por ejemplo, la función que permite realizar la descomposición de las series temporales mediante el método EMD.

# Capítulo 3

---

## Resultados, conclusiones y recomendaciones

---

### 3.1. Resultados

#### 3.1.1. Implementación del EMD en Fortran

La implementación del método EMD en Fortran se realizó siguiendo el planteamiento de la sección 2.2. En la figura 3.1 se observa que nuestra implementación genera, en promedio, un mayor número de IMFs respecto de las implementaciones de Python (python 3) y Matlab (versión R2021a), para cualquiera de las series temporales.

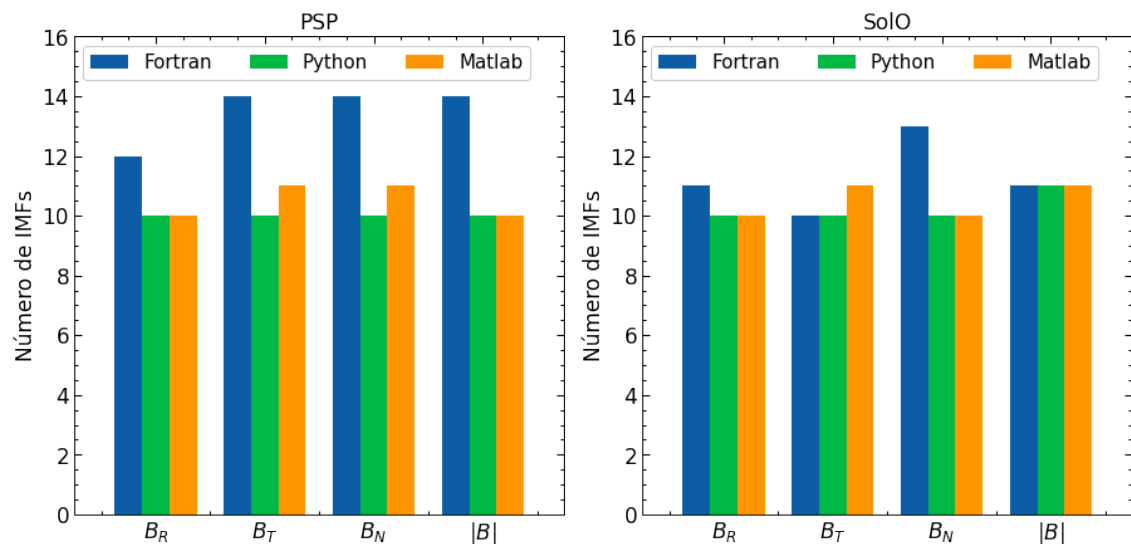


Figura 3.1: Comparación en el número de IMFs resultantes de la descomposición de las series temporales del campo magnético mediante el método EMD en Fortran, Python y Matlab. Izquierda: PSP; derecha: SolO.

Los resultados dependen de la implementación exacta del método, lo que explica la variabilidad en el número de IMFs entre las diferentes implementaciones. Las principales diferencias pueden deberse al tipo de interpolación para el cálculo de las envolventes, las condiciones de frontera de las envolventes y los criterios de detenimiento del proceso *sifting* o de toda la descomposición, como se vio en la metodología.

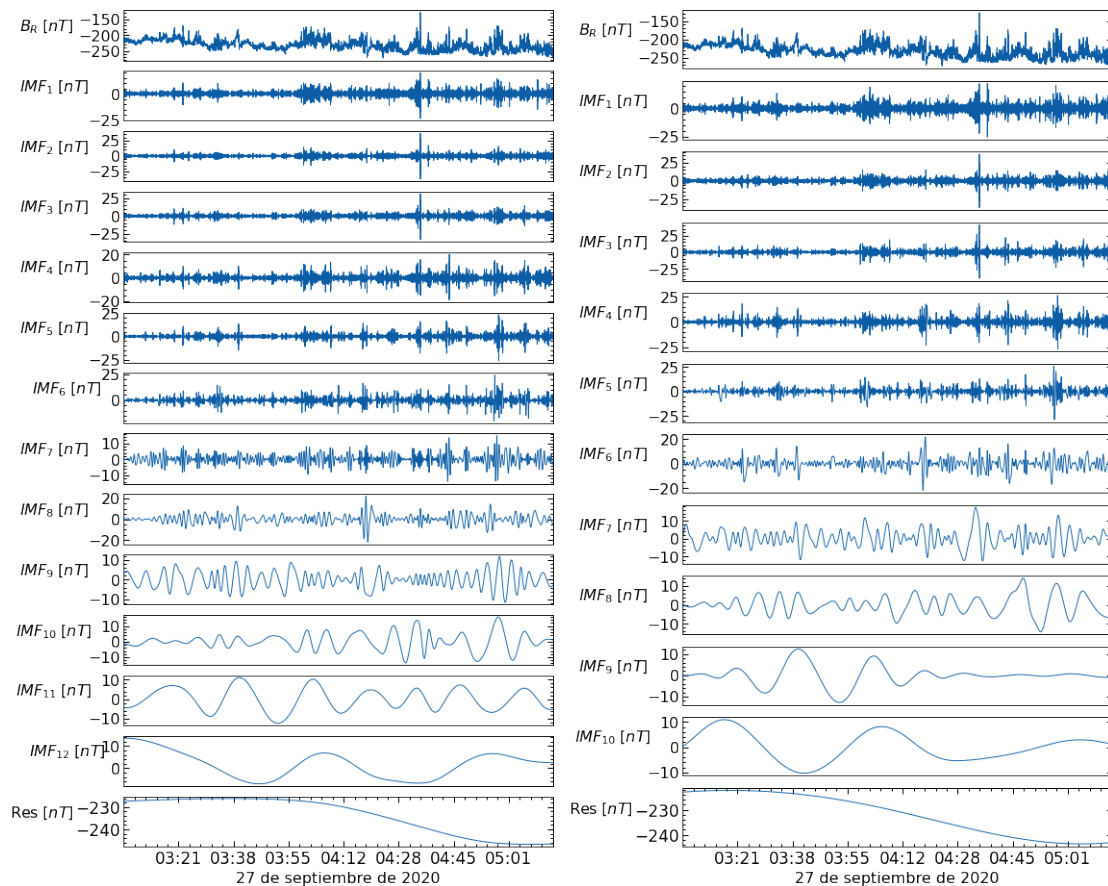


Figura 3.2: Resultado de la descomposición de  $B_R$  de PSP. De arriba hacia abajo: serie temporal de la componente radial del campo magnético; IMFs de la descomposición del método EMD y su residuo. Izquierda: Fortran, 12 IMFs; derecha: Python, 10 IMFs.

Para comparar los resultados de las implementaciones, en la figura 3.3 se muestran las reconstrucciones parciales de la serie temporal  $B_R$  de PSP, obtenidas mediante las IMFs de Python y Fortran, que se presentan en la figura 3.2. El número de IMFs que utilizan las reconstrucciones de Python y Fortran es 80 % y 75 % del total, respectivamente. Las componentes que no se incluyen son las de mayor frecuencia. Estas reconstrucciones son lo suficientemente buenas pues se parecen a la serie temporal original. La diferencia entre estas reconstrucciones es baja a excepción de puntos con un cambio muy rápido en la señal original. La razón de esto es que las implementaciones difieren un poco al extraer las características locales de las series. Además, notemos que las diferencias entre los resultados de las implementaciones son tan pequeñas, que los valores que cuantifican estas diferencias



son uno o dos órdenes de magnitud menores que los valores del campo magnético medidos en las series temporales.

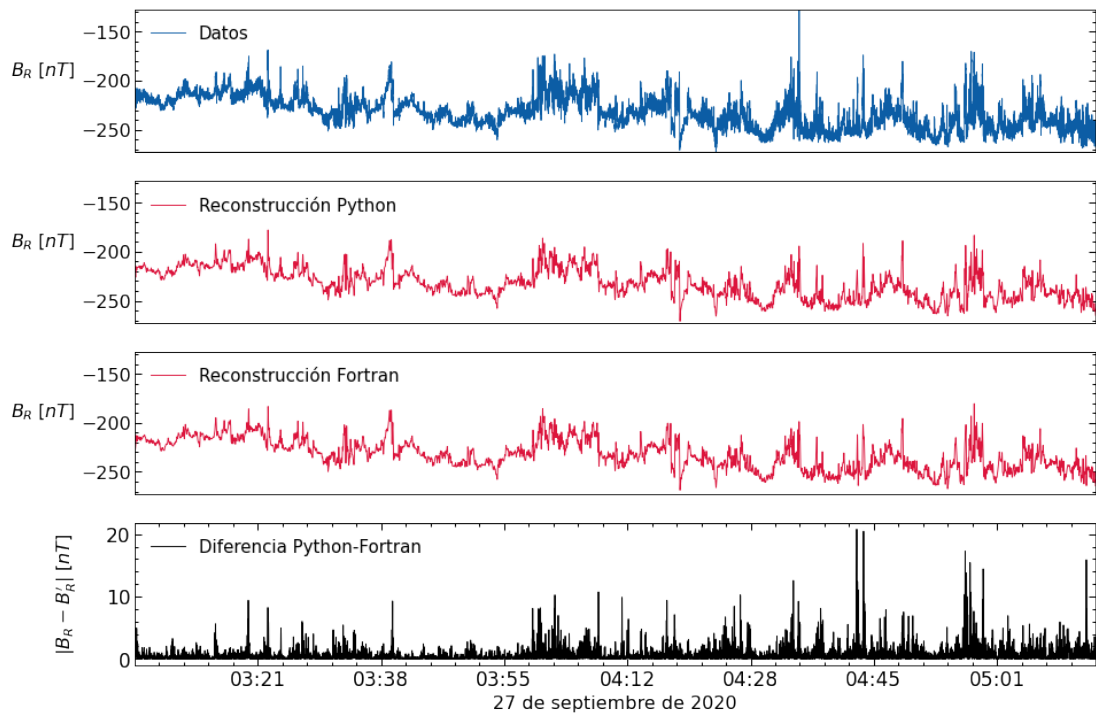


Figura 3.3: Reconstrucciones de la serie temporal  $B_R$  de PSP a partir de sus IMFs. De arriba hacia abajo: señal, reconstrucción de Python con 80 % IMFs, reconstrucción de Fortran con 75 % IMFs, valor absoluto de la diferencia entre las reconstrucciones.

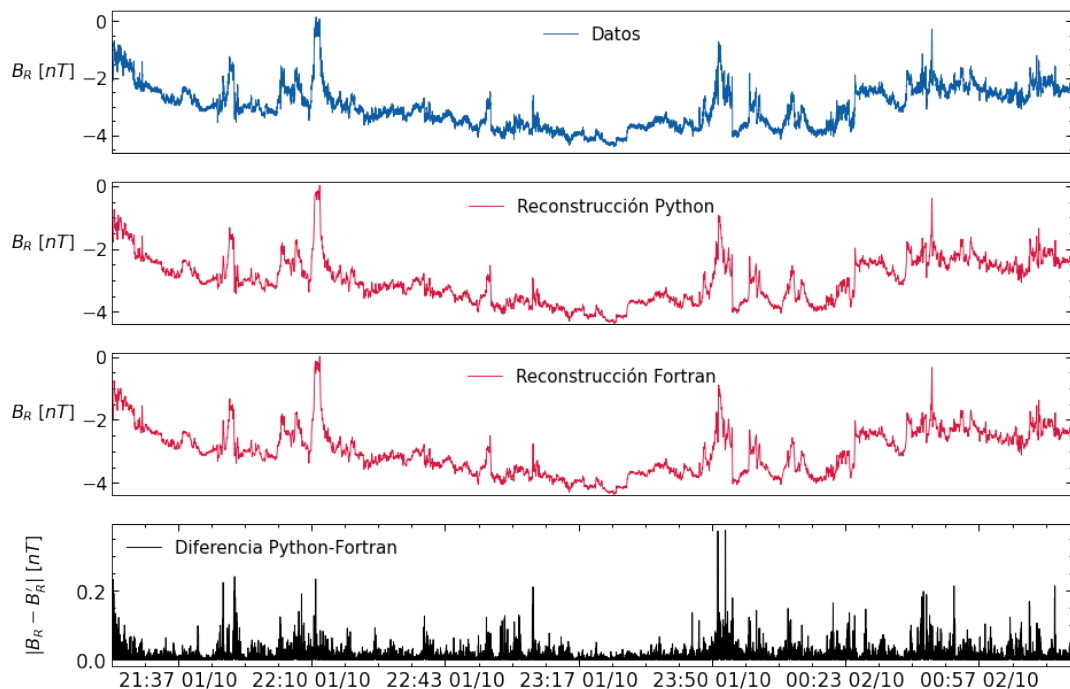


Figura 3.4: Similar a la figura 3.3 pero para los datos de SolO.

Las reconstrucciones de la serie temporal con todas sus IMFs mostraron una diferencia del orden de la precisión de los datos ( $\sim 10^{-13} - 10^{-14}$ ). Más aún, los datos originales y las diferentes reconstrucciones con todas las IMFs también dieron estos resultados. Por tanto, a pesar de que las diferentes implementaciones generan un número diferente de IMFs, los resultados entre sí son comparables. Esto valida nuestra implementación realizada en Fortran.

En la figura 3.4 se muestran las reconstrucciones parciales de la serie temporal  $B_R$  de SolO, a partir de las IMFs de menor frecuencia en la figura 3.5. Ambas reconstrucciones utilizan 8 IMFs. En este caso las reconstrucciones parciales se parecen mucho más a la serie temporal original que en los datos de PSP.

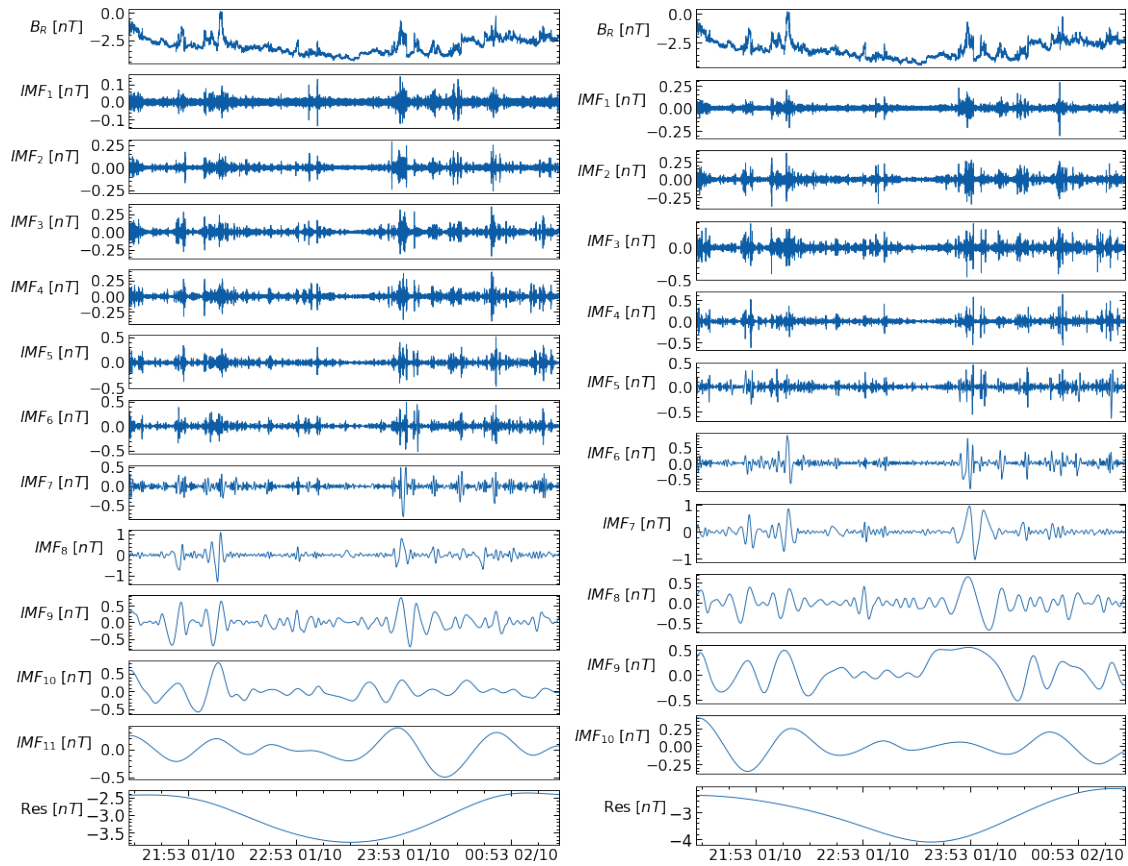


Figura 3.5: Similar a la figura 3.2 pero para los datos de SolO. Izquierda: Fortran, 11 IMFs; derecha: Python, 10 IMFs.

### 3.1.2. Correlaciones a baja frecuencia entre ventanas de tiempo de las series temporales

En esta sección presentamos los resultados de nuestro análisis exploratorio. Estos resultados se obtuvieron a través de la implementación del método EMD de Python. Resultados similares se obtuvieron con las IMFs calculadas en nuestra implementación de Fortran.

Debido al poco tiempo disponible para realizar este trabajo, solamente se pudo analizar una de las componentes del campo magnético, y se decidió que fuera la componente radial. Esta componente es predominante en las series temporales estudiadas (en promedio,  $B$  y  $B_R$  son casi paralelas) [28]. Más aún, no todas las componentes son afectadas de la misma manera por la turbulencia del viento solar, pues esta es anisotrópica [44]. A medida que crece la distancia respecto del Sol, la anisotropía de las fluctuaciones también crece [45], en relación con la evolución de la turbulencia. Sin embargo, las fluctuaciones son mayores en la dirección perpendicular al campo medio que en la dirección paralela [46]. En este caso la componente radial es la menos afectada por la turbulencia. Las razones anteriores hacen que el seguimiento de la componente radial del campo magnético sea más interesante en un primer acercamiento exploratorio, y que el análisis del resto de las componentes vaya más allá del alcance de este estudio.

El barrido se realizó con ventanas de 10 a 88 minutos en pasos de 2 minutos. El tiempo de desplazamiento ( $d$ ) también fue de 2 minutos. Por otro lado, en el barrido de las IMFs se sumaron hasta 10 componentes de las series temporales, respecto de la implementación del EMD de Python. Esto hizo que el número de operaciones necesarias para el cálculo de las correlaciones en los barridos propuestos fuera superior a  $10^4$ . Estas operaciones consumieron mucho tiempo computacional y por esta razón se omitieron las ventanas de mayor duración (que son las que tardan más tiempo en computar).

En la figura 3.6 se muestran los máximos de correlación obtenidos en los barridos de las ventanas de tiempo y las IMFs. Resalta que la tendencia de estos máximos, respecto de todas las ventanas, no varía mucho cuando se consideran más de 5 IMFs en la reconstrucción de las series temporales. Considerar menos de 4 IMFs genera correlaciones altas como consecuencia de una reducción significativa de la información de las series, por la acción del filtrado.

Si los valores máximos de una matriz de correlaciones (como la figura 2.6), calculados para una amplitud de ventana, se visualizan en un mapa de calor entonces se pueden observar patrones característicos como los de la figura 3.7. Estos *mapas de correlaciones* se obtuvieron de las reconstrucciones con las 4 IMFs de menor frecuencia, pues lo observado en la figura 3.6 sugiere que 4 es el número mínimo de IMFs que permiten obtener un buen filtrado de las series y mantener la tendencia de los máximos de correlación, respecto de las series no filtradas. Los valores de la figura 3.6 se corresponden con los máximos de cada uno de los mapas de correlaciones, es decir, aquellos obtenidos en los barridos de las ventanas de tiempo y las IMFs.

Los patrones observados indican que las series temporales contienen varios intervalos de tiempo, de diferente duración, donde su correlación es máxima. En particular, las regiones de mayor correlación para las ventanas entre 22 y 54 minutos fueron relevantes.

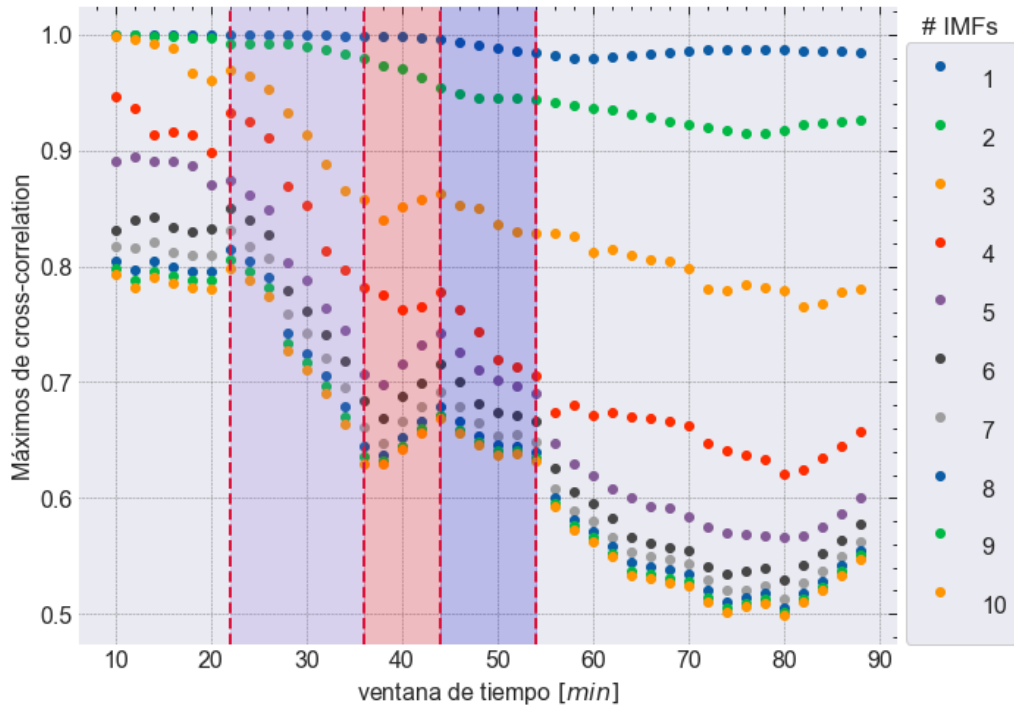


Figura 3.6: Máximos de correlación en el barrido sobre la duración de las ventanas de tiempo y el número de IMFs. Entre líneas puntadas se separan tres rangos de ventanas de tiempo para los que se observaron diferentes patrones de correlación relevantes en las series temporales de  $B_R$ .

En este rango de ventanas, los máximos de correlación de la figura 3.6 tienen gradientes pronunciados, disminuyendo rápidamente hasta los 36 minutos y con un pico en los 44 minutos, para después continuar con la tendencia decreciente. Asimismo, podemos notar que alrededor de las ventanas de 22 y 54 minutos existen cambios relativamente grandes en la correlación, para las reconstrucciones con más de 4 IMFs. Además, los intervalos de tiempo que les corresponden a este rango de ventanas se solapan con aquellos que contienen al mismo plasma espacial en las series temporales del campo magnético.

Las otras ventanas no mostraron mapas de correlaciones con regiones bien definidas y con una diferencia notoria respecto del resto del mapa. Esto lo podemos observar en los paneles 3.7(a) y 3.7(f). Además, para cada IMF, las ventanas menores a 22 minutos presentan máximos de correlación muy altos y casi no varían mucho entre sí, posiblemente debido a que se consideran pocos datos para hacer las cuentas. En el caso de ventanas de más de 54 minutos puede que la semejanza entre las series temporales disminuya y por eso menos intervalos de tiempo muestran alta correlación. En las ventanas de más de 80 minutos se observa un incremento en el máximo de la correlación, aunque los mapas de correlación muestran una única región poco localizada, es decir, valores altos muy dispersos. Esto simplemente reduce los posibles intervalos de tiempo que se pueden relacionar con la presencia del mismo plasma.

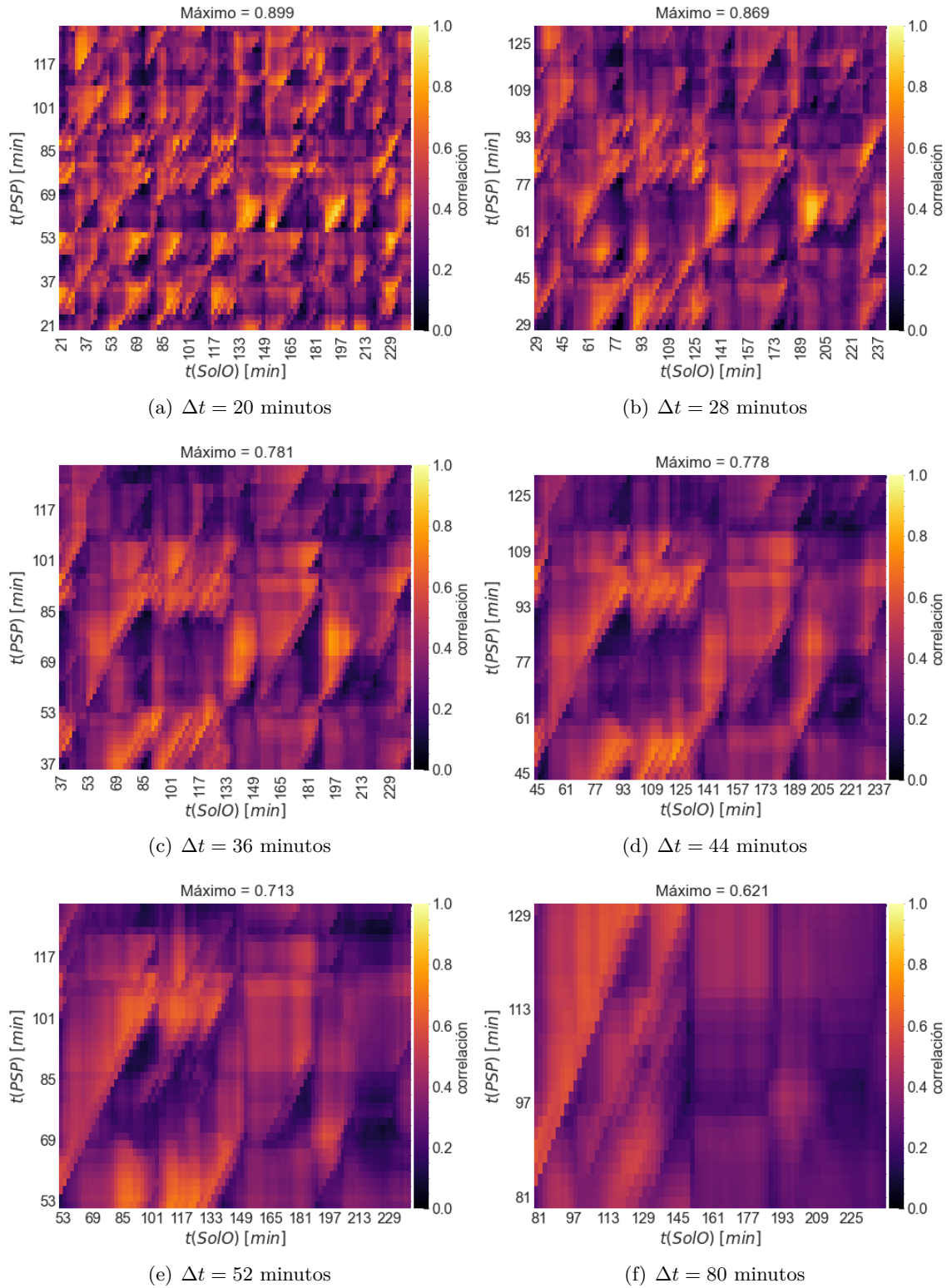


Figura 3.7: Mapas de correlaciones para algunas ventanas de tiempo en las series temporales de  $B_R$  de PSP y SolO. Las correlaciones se calcularon entre datos reconstruidos con 4 IMFs de baja frecuencia. De arriba hacia abajo y de izquierda a derecha: ventanas de tiempo de 20, 28, 36, 44, 52 y 80 minutos.

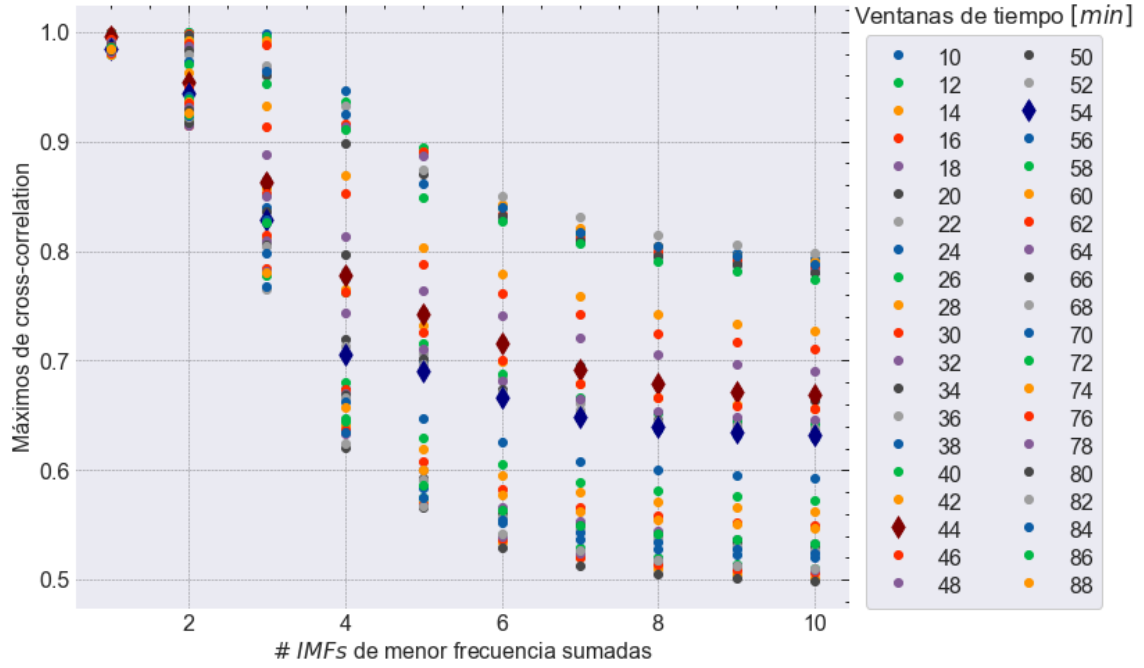


Figura 3.8: Máximos de correlación respecto del número de IMFs de menor frecuencia que reconstruyen las ventanas de tiempo de las series temporales de  $B_R$ . Las ventanas de 44 y 54 minutos se indican con marcadores diferentes como ejemplo de la tendencia.

Las tres bandas de ventanas sombreadas de la figura 3.6 se relacionan con las regiones de correlación relevantes de la figura 3.7. Sólo las bandas donde la correlación disminuye muestran dos regiones localizadas. La banda central es algo así como una transición entre las regiones de las otras bandas. A pesar de la alta correlación de las regiones observadas, no es la alta correlación lo relevante en este estudio. Los cambios significativos en la correlación a medida que se añaden más IMFs son de interés. Quizá las ventanas donde existe un gran salto en la correlación, ya sea para un número determinado de IMFs o al variar las IMFs, podrían ser físicamente significativas. Por ejemplo, en la figura 3.6, las reconstrucciones con 4 IMFs destacan estos saltos para las ventanas de 22 y 54 minutos.

Los expertos suelen utilizar series temporales cuya duración es superior a los 60 minutos para sus estudios [28]. La razón de esto es que para tiempos de menor duración la turbulencia tiende a destruir las estructuras de menor tamaño o provoca que la correlación entre estas (estructuras) disminuya mucho durante la expansión del viento solar [13]. Sin embargo, los tiempos encontrados ( $\sim 22 - 54$  minutos), con este análisis de carácter puramente exploratorio, podrían tener alguna relevancia física, más allá de la relación con los intervalos de tiempo que contienen al mismo plasma espacial, y por tanto se deberían estudiar más detenidamente para descubrir su importancia.

En las figuras 3.8 y 3.9 se muestran resultados similares a lo observado en la figura 3.6. El máximo de correlación no varía mucho para las reconstrucciones con más de 5

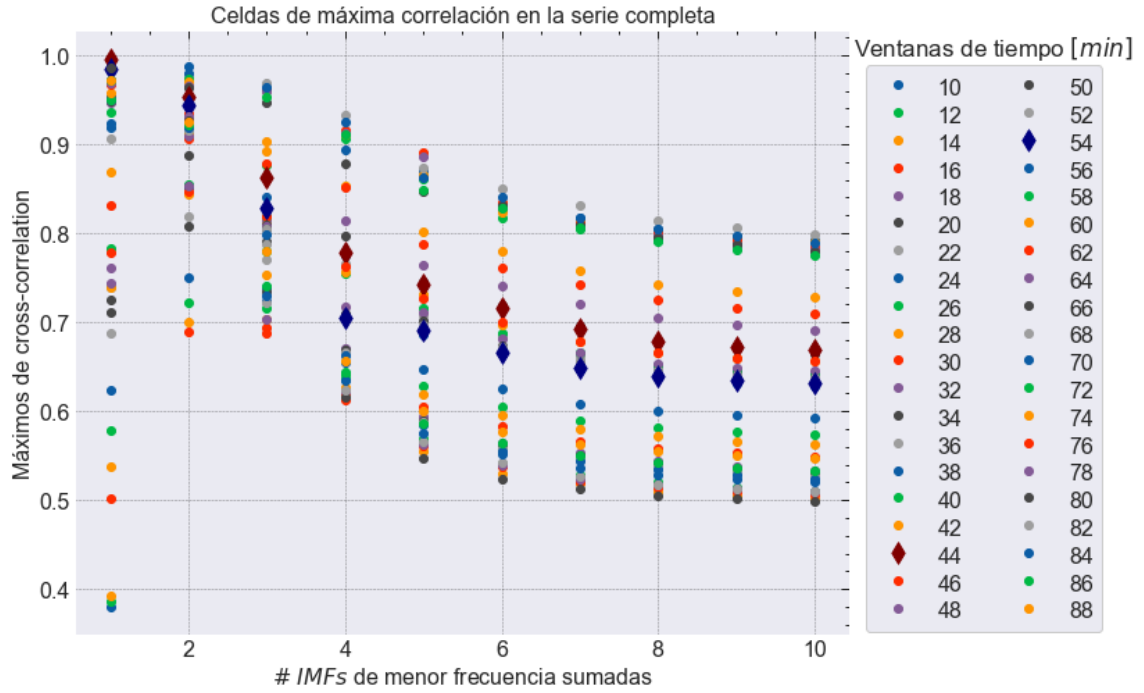


Figura 3.9: Similar a 3.8 pero para las particiones de mayor correlación que se obtienen con la serie temporal completa, es decir, las particiones son fijas en el barrido.

IMFs. Además, en la figura 3.8 se observa que el máximo de correlación en cada uno de los mapas se incrementa al reducir el número de IMFs de las señales y que disminuye al considerar ventanas más grandes. Por otro lado, la figura 3.9 indica que los intervalos de tiempo más correlacionados, con 10 IMFs (cuando no se han filtrado las altas frecuencias), no siempre incrementan su correlación al reducir el número de IMFs en la reconstrucción de las señales.

La reconstrucción de las series temporales de  $B_R$  con 5 IMFs mostró una frecuencia característica (frecuencia en el valor máximo del espectro de potencias de Fourier justo antes del comportamiento de ley de potencia en la curva  $\log - \log$ ; el lector puede dirigirse a las referencias para más detalles, en particular [32]) de  $9 \times 10^{-3} Hz$  para PSP y  $1.66 \times 10^{-3} Hz$  para SolO. Estos valores son cercanos a la frecuencia característica de la quinta IMF de menor frecuencia,  $1 \times 10^{-2} Hz$  para PSP y  $4.16 \times 10^{-3} Hz$  para SolO, lo cual es esperado por la acción del filtro. Estos valores de frecuencia podrían sugerir una escala característica para identificar cuándo las partes no turbulentas de dos series temporales del campo magnético del viento solar están altamente correlacionadas y quizá verificar si pertenecen al mismo plasma. Sin embargo, mayores detalles sobre las implicaciones físicas no son del todo claros y se necesita un mayor estudio.

En las figuras 3.10 y 3.11 se presenta el caso particular de la reconstrucción de las series temporales (de PSP y SolO) con 4 IMFs de baja frecuencia, para dos ventanas de tiempo de 54 minutos. Se observa que las reconstrucciones se ajustan suavemente al comportamiento

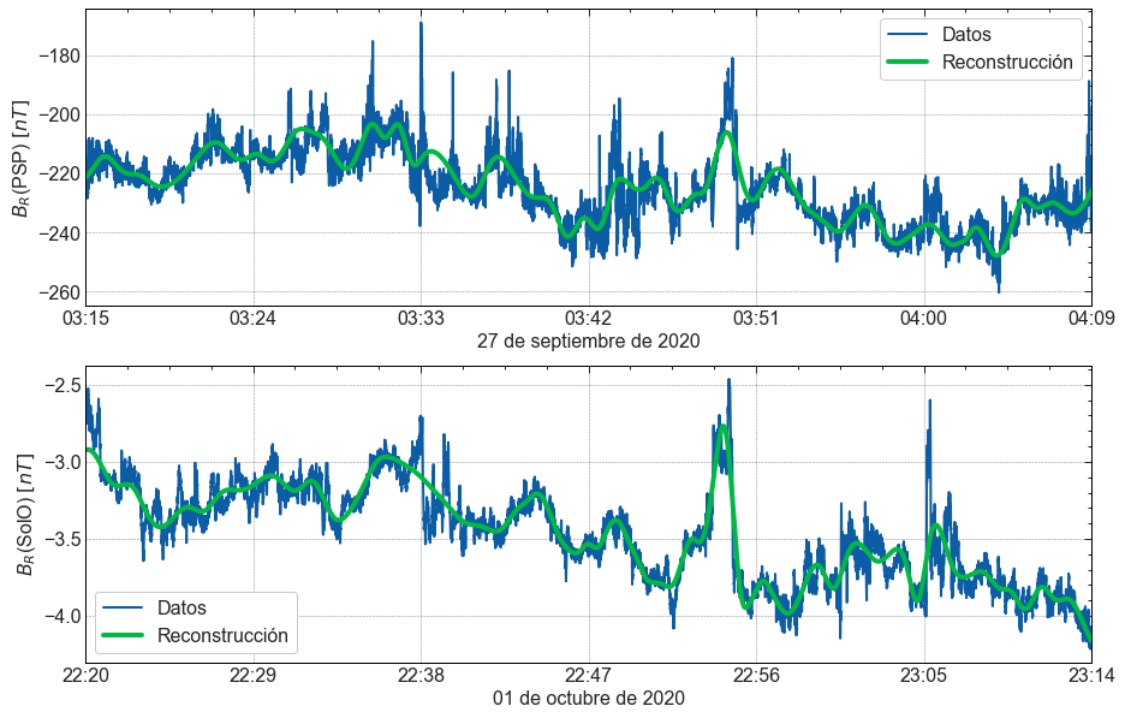


Figura 3.10: Reconstrucción de las series temporales de  $B_R$  con las 4 IMFs de menor frecuencia en una ventana de 54 minutos. Arriba: datos de PSP en el intervalo de tiempo 03:15-04:09 UT el 27 de septiembre de 2020; abajo: datos de SolO en el intervalo de tiempo 22:20-23:14 UT el 01 de octubre de 2020.

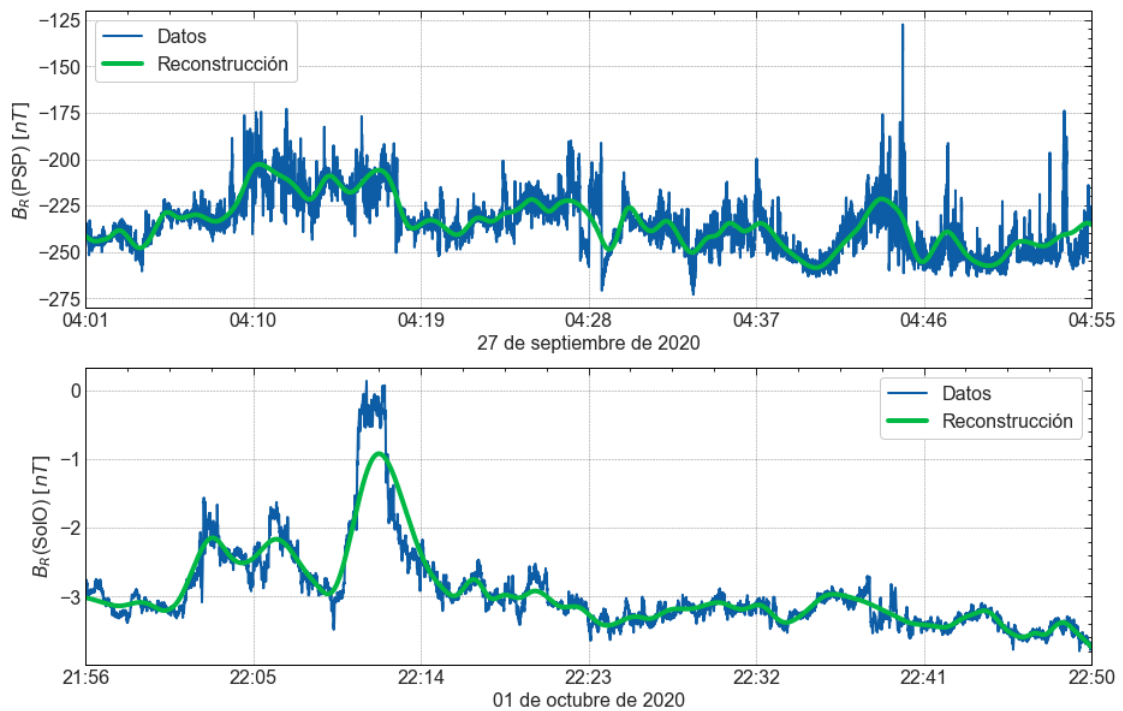


Figura 3.11: Similar a 3.10. Arriba: datos de PSP en el intervalo de tiempo 04:01-04:55 UT el 27 de septiembre de 2020; abajo: datos de SolO en el intervalo de tiempo 21:56-22:50 UT el 01 de octubre de 2020.



de los datos. El valor máximo de la función cross-correlation entre las señales reconstruidas es de 0.705 para la figura 3.10 y 0.656 para la figura 3.11. Los intervalos de tiempo están relacionados con cada una de las regiones relevantes en los mapas de correlación, teniendo los pares de ventanas más correlacionados para 54 minutos. Estos intervalos son entre las horas 03:15 - 04:09 UT del 27 de septiembre de 2020 en PSP y 22:20 - 23:14 UT del 1 de octubre de 2020 en SolO. Por otro lado, los intervalos de tiempo de las otras reconstrucciones son entre las horas 04:01 - 04:55 UT en PSP y 21:56 - 22:50 UT en SolO, en los días respectivos.

Las series temporales analizadas en este trabajo se corresponden con los intervalos de tiempo ampliados que se estudiaron en la investigación que se toma de referencia [28]. En otras palabras, no se asegura que toda la duración de las señales pertenece al mismo plasma espacial, sino solamente una parte. Los intervalos de tiempo encontrados, en las regiones de alta correlación, están contenidos o se solapan con aquellos hallados en el estudio anterior, indicando nuevamente que los intervalos reportados son los que contienen la información del mismo plasma espacial.

## 3.2. Conclusiones y recomendaciones

Nuestra implementación del método EMD en Fortran produjo, en promedio, un mayor número de IMFs respecto de las implementaciones profesionales en Matlab y Python. Esta diferencia se debe al grado de optimización de las implementaciones. Sin embargo, los resultados fueron comparables dado que la reconstrucción de las series temporales fue similar. Esto permitió validar la implementación hecha en Fortran.

El cálculo de las correlaciones propuestas se limitó a la componente radial del campo magnético. En el barrido sobre el número de IMFs se estimó que alrededor de 5 IMFs, con la implementación de Python, permiten reconstruir las series de forma adecuada, realizar un buen filtrado de las altas frecuencias y mantener la tendencia de los máximos de correlación para las distintas amplitudes de ventana, con respecto a las series no filtradas.

Por otro lado, el barrido de las ventanas de tiempo mostró que ventanas entre 22 y 54 minutos presentan una alta correlación (3.7) y que sus valores máximos tienen gradientes pronunciados (3.6), que los diferencian de ventanas menores a 22 minutos o mayores a 54 minutos. Esto sugiere que las amplitudes de ventana entre 22 y 54 minutos podrían ser físicamente significativas. Sin embargo, este estudio, que tiene un carácter puramente exploratorio, sólo permitió identificar estas amplitudes, pero no se pudo concluir nada sobre sus implicaciones físicas pues los detalles no son del todo claros. Por último, los intervalos de tiempo, asociados a estas amplitudes de ventana, se solapan con aquellos reportados en el estudio anterior [28] sobre los datos analizados en este trabajo, es decir,

se relacionan con la información del mismo plasma a distintas distancias.

Finalmente, se recomienda estudiar las propiedades físicas, de la parte no turbulenta de las series temporales, de las ventanas de tiempo entre 22 y 54 minutos para su determinar su relevancia y sus posibles implicaciones físicas. También, un trabajo que siguiera la línea iniciada en este estudio debería investigar las frecuencias características de las reconstrucciones de las series con 5 IMFs y buscar si existe algún significado físico. De igual manera, se debe continuar explorando sobre las otras componentes del campo magnético y revisar el número de IMFs para obtener un buen filtro de las altas frecuencias.

# Capítulo A

---

## Anexos

---

### A.1. Código en Fortran con el método EMD

```
1  !=====
2  !Module containing subroutines to perform the emd algorithm
3  !=====
4  module emd_mod
5      implicit none
6      integer, parameter :: dp = selected_real_kind(15, 307)
7      contains
8  !-----
9  !Adds an element to an array
10 !-----
11 subroutine append(array, element)
12     integer :: i, insize
13     real(dp), intent(in) :: element
14     real(dp), dimension(:), allocatable, intent(inout) :: array
15     real(dp), dimension(:), allocatable :: outlist
16
17     if(allocated(array)) then
18         insize = size(array)
19         allocate(outlist(insize+1))
20         do i=1,insize
21             outlist(i) = array(i)
22         end do
23         outlist(insize+1) = element
24         deallocate(array)
25         call move_alloc(outlist, array)
26     else
27         allocate(array(1))
28         array(1) = element
29     end if
30 end subroutine append
31 !-----
32 !Finds the local maxima and minima of an array and its positions in x-axis
33 !-----
34 subroutine findLocalMaximaMinima(x, array, x_minima, minima, x_maxima, maxima)
35     integer :: N, i, m, num_minima, num_maxima
36     real(dp), intent(in) :: x(:), array(:)
```

```

37 real(dp), dimension(:), intent(inout), allocatable :: x_minima, minima, &
38     x_maxima, maxima
39 real(dp), dimension(:), allocatable :: x_aux, y_aux
40
41 allocate(x_minima(0), minima(0), x_maxima(0), maxima(0))
42 N = size(array)
43 num_minima = 0
44 num_maxima = 0
45 do i=2, N-1
46     !condition for local minima
47     if((array(i-1)>array(i)).and.(array(i)<array(i+1))) then
48         call append(x_minima, x(i))
49         call append(minima, array(i))
50         num_minima = num_minima + 1
51     !condition for local maxima
52     else if((array(i-1)<array(i)).and.(array(i)>array(i+1))) then
53         call append(x_maxima, x(i))
54         call append(maxima, array(i))
55         num_maxima = num_maxima + 1
56     end if
57 end do
58
59 !Boundary conditions to perform Cubic Splines
60 !This adds 2 values to local maxima and minima: 4 in total
61 !To get the exact number, the size of each must be reduced by 2
62 if((num_minima .ne. 0).and.(num_maxima .ne. 0)) then
63     !boundary conditions for minima
64     m = size(x_minima)
65     allocate(x_aux(m+2), y_aux(m+2))
66     x_aux(1) = x(1)
67     x_aux(2:m+1) = x_minima(1:m)
68     x_aux(m+2) = x(N)
69     y_aux(1) = minima(1)
70     y_aux(2:m+1) = minima(1:m)
71     y_aux(m+2) = minima(m)
72     deallocate(x_minima, minima)
73     allocate(x_minima(m+2), minima(m+2))
74     x_minima = x_aux
75     minima = y_aux
76     deallocate(x_aux, y_aux)
77
78     !boundary conditions for maxima
79     m = size(x_maxima)
80     allocate(x_aux(m+2), y_aux(m+2))
81     x_aux(1) = x(1)
82     x_aux(2:m+1) = x_maxima(1:m)
83     x_aux(m+2) = x(N)
84     y_aux(1) = maxima(1)
85     y_aux(2:m+1) = maxima(1:m)
86     y_aux(m+2) = maxima(m)
87     deallocate(x_maxima, maxima)
88     allocate(x_maxima(m+2), maxima(m+2))
89     x_maxima = x_aux
90     maxima = y_aux
91     deallocate(x_aux, y_aux)
92 end if
93 end subroutine
94 !-----
95 !Performs Cubic Splines using 'x', 'y' and 'x_splines' vectors as input and
96 !returns 'y_splines' vector as output, interpolating the data in 'y'
97 !-----

```

```

98 subroutine cubic_splines(x, y, x_splines, y_splines)
99   real(dp), dimension(:), intent(in) :: x, y, x_splines
100  real(dp), dimension(:), intent(inout) :: y_splines
101  real(dp), dimension(:), allocatable :: b, h, c, d, l, dd, u
102  integer :: i, j, k, Nnodes, N_splines
103  real(dp) :: xinter, yinter
104
105  Nnodes = size(x)
106  N_splines = size(x_splines)
107  allocate(b(Nnodes), h(Nnodes-1), c(Nnodes-1), d(Nnodes-1), l(Nnodes), &
108          dd(Nnodes), u(Nnodes))
109
110  !Tridiagonal matrix and vector of coefficients for the method
111  !A(1, dd, u)x = b
112  do i = 1, Nnodes-1
113     h(i) = x(i+1)-x(i)
114     l(i+1) = 1.0_dp/h(i)
115     u(i) = 1.0_dp/h(i)
116     if(i>1) then
117        dd(i) = 1./h(i-1) + 1.0_dp/h(i)
118        b(i) = y(i+1)/(h(i)*h(i)) + y(i)*(1./(h(i-1)*h(i-1))-1./(h(i)*h(i))) - &
119            y(i-1)/(h(i-1)*h(i-1))
120     end if
121  end do !i
122  dd(1) = 1.0_dp/h(1)
123  dd(Nnodes) = 1.0_dp/h(Nnodes-1)
124  dd(1:Nnodes) = 2.0_dp*dd(1:Nnodes)
125  b(1) = (y(2)-y(1))/(h(1)*h(1))
126  b(Nnodes) = (y(Nnodes)-y(Nnodes-1))/(h(Nnodes-1)*h(Nnodes-1))
127  b(1:Nnodes) = 3.0_dp*b(1:Nnodes)
128
129  !Solution of the linear system
130  do i=2,Nnodes
131     l(i) = l(i)/dd(i-1)
132     dd(i) = dd(i) - l(i)*u(i-1)
133     b(i) = b(i)-l(i)*b(i-1)
134  end do !i
135  b(Nnodes) = b(Nnodes)/dd(Nnodes)
136  do i = Nnodes-1, 1, -1
137     b(i) = (b(i) - u(i)*b(i+1))/dd(i)
138  end do !i
139
140  !Cubic Splines Interpolation
141  do i = 1, Nnodes-1
142     c(i) = 3.0_dp*(y(i+1)-y(i))/(h(i)*h(i))-2.0_dp*b(i)/h(i)-b(i+1)/h(i)
143     d(i) = 2.0_dp*(y(i)-y(i+1))/(h(i)**3) + (b(i)+b(i+1))/(h(i)*h(i))
144  end do
145  j = 1
146  do i = 1, N_splines
147     xinter = x_splines(i)
148     do k = 1, Nnodes-1
149        if((x(k)<xinter).and.(xinter < x(k+1))) j = k
150     end do !k
151     yinter = y(j) + b(j)*(xinter-x(j)) + c(j)*(xinter-x(j))*(xinter-x(j)) + &
152        d(j)*(xinter-x(j))**3
153     y_splines(i) = yinter
154  end do !i
155  deallocate(h,b,c,d,l,dd,u)
156 end subroutine
157 !
158 !Performs the main steps in the sifting process.

```

```

159 !It takes 'x' and 'y' as input vectors and returns a proto-IMF 'hk' and also
160 !'mean' and 'diff_extremas_zeros' in order to check if 'hk' is an IMF.
161 !-----
162 subroutine sifting_steps(x, y, hk, mean, diff_extremas_zeros)
163   real(dp), dimension(:), intent(in) :: x, y
164   real(dp), dimension(:), intent(inout) :: hk
165   real(dp), dimension(:), allocatable :: x_minima, minima, x_maxima, maxima, &
166     ys_low, ys_up, ys_mean
167   integer, intent(inout) :: diff_extremas_zeros
168   real(dp), intent(inout) :: mean
169   integer :: N_file, i, number_zero_crossings
170
171   N_file = size(x)
172   call findLocalMaximaMinima(x, y, x_minima, minima, x_maxima, maxima)
173   allocate(ys_low(N_file))
174   call cubic_splines(x_minima, minima, x, ys_low)
175   allocate(ys_up(N_file))
176   call cubic_splines(x_maxima, maxima, x, ys_up)
177   allocate(ys_mean(N_file))
178   ys_mean = 0.5*(ys_low + ys_up)
179
180   hk = y - ys_mean
181   mean = 0.0_dp
182   do i = 1, N_file
183     mean = mean + hk(i)
184   end do
185   mean = mean/N_file
186
187   number_zero_crossings = 0
188   do i = 1, N_file - 1
189     if(hk(i)*hk(i+1) < 0) number_zero_crossings = number_zero_crossings + 1
190   end do
191
192   diff_extremas_zeros = abs(size(maxima)+size(minima)-number_zero_crossings - 4)
193   !We subtract 4 to correct the size of maxima and minima
194   deallocate(x_minima, minima, x_maxima, maxima, ys_low, ys_up, ys_mean)
195 end subroutine
196 !-----
197 !Standard Deviation criterion to stop the sifting process
198 !-----
199 subroutine stopping_criterion1(hkm1, hk, SD)
200   real(dp), dimension(:), intent(in) :: hkm1, hk
201   real(dp), intent(inout) :: SD
202   integer :: i, total_time
203   real(dp), parameter :: eps=1e-8
204
205   total_time = size(hk)
206   SD = 0.0_dp
207   do i = 1, total_time
208     SD = SD + abs(hk(i)-hkm1(i))*abs(hk(i)-hkm1(i))/(hkm1(i)*hkm1(i)+eps)
209   end do
210 end subroutine
211 !-----
212 !Evaluates if the input parameters guarantee an IMF
213 !-----
214 subroutine is_imf(param1, param2, success_imf)
215   !Difference between the number of extremas and zero crossings
216   integer, intent(in) :: param1
217   real(dp), intent(in) :: param2 !mean of Proto-IMFs
218   logical, intent(inout) :: success_imf
219   real(dp), parameter :: eps_mean = 4e-2 !chosen mean tolerance

```

```

220
221     if((param1 <= 1) .and. (abs(param2) <= eps_mean)) then
222         success_imf = .true.
223     else
224         success_imf = .false.
225     end if
226 end subroutine
227 !-----
228 !Evaluates the number of extremas of the residue to stop the decomposition
229 !-----
230 subroutine residual_condition(x, r, num_residual_extremas)
231     real(dp), dimension(:), intent(in) :: x, r
232     integer, intent(inout) :: num_residual_extremas
233     real(dp), dimension(:), allocatable :: rx_minima, r_minima, rx_maxima, r_maxima
234     integer :: size_minima, size_maxima
235
236     !Precision correction in the residual values.
237     !This avoids getting wrong IMFs.
238     call findLocalMaximaMinima(x, anint(r*1e8)/1e8, rx_minima, r_minima, &
239         rx_maxima, r_maxima)
240     size_minima = size(r_minima)
241     size_maxima = size(r_maxima)
242     !Size correction for minima and maxima
243     if((size_minima.ne.0).and.(size_maxima.ne.0)) then
244         num_residual_extremas = size_minima + size_maxima - 4
245     else
246         num_residual_extremas = 0
247     end if
248     deallocate(rx_minima, r_minima, rx_maxima, r_maxima)
249 end subroutine
250 end module
251 !-----
252 !Main program to perform EMD method on magnetic time series.
253 !-----
254 program emd
255     use emd_mod
256     implicit none
257
258     !MAX_IMFs~log2(N): in this case MAX_IMFs~ 16, 17: expected number of IMFs
259     !MAX_PMFs: maximum number of iterations to perform the sifting process
260     integer, parameter :: MAX_IMFs=16 + 2, MAX_PMFs=20
261     real, parameter :: SD_threshold = 0.3_dp
262     integer :: io1=1, io2=2
263     integer :: N_file
264     real(dp), dimension(:), allocatable :: x, y, hkml, hk, rj, &
265         time, Br, Bn, Bt, B
266     integer :: i, diff_extremas_zeros, num_residual_extremas, num_IMFs, it_sift
267     character(len=80) :: nfout
268     real(dp) :: mean, SD
269     logical :: success_imf
270
271     !size of time series data, name of output file with IMFs
272     N_file = 62400 !PSP
273     nfout = 'imfs_PSP_Br.dat'
274     !N_file = 115201 !SolO
275     !nfout = 'imfs_SolO_B.dat'
276     nfout = trim(nfout)
277
278     allocate(x(N_file), y(N_file), hk(N_file), hkml(N_file), rj(N_file), &
279         time(N_file), Br(N_file), Bt(N_file), Bn(N_file), B(N_file))
280

```

```

281 !data reading
282 open(unit=io1, file='pspr_20200927_0305_20200927_0515.txt', action='read')
283 !open(unit=io1, file='solo_20201001_2120_20201002_0120.txt', action='read')
284 do i = 1, N_file
285     read(io1, *) time(i), Br(i), Bt(i), Bn(i), B(i)
286 end do
287 close(io1)
288
289 !variables on which the EMD method will be applied
290 x(1:N_file) = time(1:N_file)/8.0_dp !sample rate = 8Hz
291 y(1:N_file) = Br(1:N_file)
292
293 open(unit=io2, file=nfout, status='unknown', action='write')
294
295 !Residual condition: monotonous or constant behavior: at most 1 extrema
296 num_residual_extremas = 2
297 num_IMFs = 0
298 rj = y !initial condition for residue
299 !-----EMD-----
300 do while((num_residual_extremas > 1).and.(num_IMFs < MAX_IMFs))
301     num_IMFs = num_IMFs + 1
302     print*, 'IMF: ', num_IMFs
303     hkml = rj
304
305 !-----SIFTING-----
306 SD = 1.0_dp
307 it_sift = 0
308 do while((SD > SD_threshold).and.(it_sift < MAX_PMFs))
309     it_sift = it_sift + 1
310     hk = 0.0_dp
311     call sifting_steps(x, hkml, hk, mean, diff_extremas_zeros)
312     call is_imf(diff_extremas_zeros, mean, success_imf)
313     call stopping_criterion1(hkml, hk, SD)
314     !condition to stop sifting
315     if(success_imf.eqv(.true.)) exit
316     hkml = hk
317 end do !end of sifting process
318
319 !report of sifting results
320 write(*, "(A5, I2, A10, I5, A8, E20.8, A7, L1, A12, F20.8)" &
321     '#it: ', it_sift, '| param1: ', diff_extremas_zeros, &
322     '| mean: ', mean, '| imf: ', success_imf, '| criterion: ', SD)
323
324 if(it_sift == MAX_PMFs) then
325     print*, 'Approximated sifting'
326 else if (SD < 0.3_dp) then
327     print*, 'SD condition achieved'
328 end if
329
330 !residue update
331 rj = rj - hk
332 write(io2, *) hk
333 call residual_condition(x, rj, num_residual_extremas)
334 print*, '|-----Extrema of residue-----|', num_residual_extremas
335 end do !end of EMD
336
337 write(io2, *) rj
338
339 print*, 'Extrema of residue: ', num_residual_extremas
340 print*, 'Number of IMFS: ', num_IMFs
341

```



```

342 deallocate(x, y, hkml, hk, rj, time, Br, Bn, Bt, B)
343 close(io2)
344 end program
345 !

```

---

## A.2. Código en Python para realizar el barrido sobre las ventanas de tiempo y el número de IMFs

```

=====
#Program to perform the cross-correlation scan on the time windows and on
#the number of IMFS
=====
#libraries
import numpy as np
import emd as emd2
import numba
from numba import jit
import seaborn as sns; sns.set_theme()

#data
psp_data = np.loadtxt('pspr_20200927_0305_20200927_0515.txt')
psp_data = np.transpose(psp_data)
psp_data[0] = psp_data[0]/8.
solo_data = np.loadtxt('solo_20201001_2120_20201002_0120.txt')
solo_data = np.transpose(solo_data)
solo_data[0] = solo_data[0]/8.
#-----
#Functions
#-----
#Cross-Correlation function
@numba.jit
def cross_corr(x, y):
    n = len(x)
    mean_x = np.mean(x)
    mean_y = np.mean(y)
    std_x = np.std(x)
    std_y = np.std(y)

    xcorr = []
    for k in range(n):
        suma = 0.
        for i in range(n-k):
            suma += (x[i]-mean_x)*(y[i+k]-mean_y)
        xcorr.append(suma)
    xcorr = np.array(xcorr)/(n*std_x*std_y)
    return xcorr

#Function to get the partitions of PSP and Solo (portion of the signals)
#of the correlation values from the maximum to the minimum
def get_jixcorrs(W):
    length = np.size(W)
    aux = np.reshape(W, (1, length))
    aux2 = np.sort(aux, kind='quicksort')
    aux3 = [np.where(W == aux2[0][i]) for i in range(length)]

```

```

indexs = []
for k in range(length):
    i, = aux3[k][1]
    j, = aux3[k][0]
    indexs.append([j, i, W[j][i]])
indexs = indexs[::-1]
return indexs
#Funtion to perform the cross-correlation scan on the number of IMFS
def params_xcorr(field_component, time_min, delay_min):
    #EMD parameters
    config = emd2.sift.get_config('sift')
    config['imf_opts/sd_thresh'] = 0.3

    #parameters
    T1 = 62400 #130 min PSP
    T2 = 115201 #240 min Solo
    time_width = time_min*60*8
    delay = delay_min*60*8 #time to change from one partition to another
    n_inter1 = int(np.floor((T1-time_width)/delay)) + 1
    n_inter2 = int(np.floor((T2-time_width)/delay)) + 1
    Tp1 = time_width + delay*(n_inter1-1) #<=T1
    Tp2 = time_width + delay*(n_inter2-1) #<=T2

    #data
    x1 = psp_data[field_component][:Tp1]
    x2 = solo_data[field_component][:Tp2]

    #EMD
    imf1 = emd2.sift.sift(x1) #Residue included
    imf2 = emd2.sift.sift(x2)
    num_imfs1 = imf1.shape[1]-1
    num_imfs2 = imf2.shape[1]-1
    start_index_imfs1 = [i for i in range(num_imfs1)]
    start_index_imfs2 = [i for i in range(num_imfs2)]

    #SCAN
    params = []
    print('Time window: {} minutes'.format(time_min))
    sum_imfs = min(start_index_imfs1[-1], start_index_imfs2[-1])
    for k in range(sum_imfs):
        print('# imfs added : PSP({}), Solo({})'.format(num_imfs1-k, num_imfs2-k))

    #Elimination of high frequencies
    y1 = np.sum(imf1[:, start_index_imfs1[k] : ], axis=1)
    y2 = np.sum(imf2[:, start_index_imfs2[k] : ], axis=1)

    #cross-correlation between partitions over the time series
    #scan for a particular time window
    xcorrs = []
    max_lags = []
    for i in range(n_inter1):
        start1 = i*delay
        end1 = time_width + i*delay
        for j in range(n_inter2):
            start2 = j*delay
            end2 = time_width + j*delay
            aux = cross_corr(y1[start1:end1], y2[start2:end2])
            index, = np.where(aux == np.max(aux))
            xcorrs.append(np.max(aux))
            max_lags.append(index[0])

```

```

xcorr_map = np.reshape(xcorr, (n_inter1, n_inter2))
xcorr_lags = np.reshape(max_lags, (n_inter1, n_inter2))
max_vals = get_jixcorr(xcorr_map)

#map of maximum correlations
ticks_step = 8
xlabel0 = np.arange(time_min+1, time_min+delay_min*(n_inter2-1), delay_min,
↳ dtype=int)
ylabel0 = np.arange(time_min+1, time_min+delay_min*(n_inter1-1), delay_min,
↳ dtype=int)
xlabel1 = np.char.mod('%d', xlabel0)
ylabel1 = np.char.mod('%d', ylabel0)
xticks = np.arange(0, n_inter2, 1)
yticks = np.arange(0, n_inter1, 1)

fig, ax = plt.subplots(figsize=(width, height))
ax = sns.heatmap(xcorr_map, vmin=0, vmax=1, cmap='inferno',
  cbar_kws={'label': "correlation", 'pad': 0.005}, annot_kws={"size": 17})
plt.text(0.35, 0.9, "Maximum = {:.3f}".format(np.max(xcorr_map)), fontsize=17,
  transform=plt.gcf().transFigure)
ax.invert_yaxis()
ax.set_xlabel(r"$t(SOLO) \ ; [min]$", fontsize=17)
ax.set_ylabel(r"$t(PSP) \ ; [min]$", fontsize=17)

ax.set_yticks(yticks[1::ticks_step])
ax.set_yticklabels(ylabel1[1::ticks_step])
ax.set_xticks(xticks[1::ticks_step])
ax.set_xticklabels(xlabel1[1::ticks_step])
plt.yticks(rotation=0)
#save map of maximum correlations in img folder
plt.savefig('img/fig-' + str(num_imfs1 - start_index_imfs1[k]) + '&' +
  str(num_imfs2 - start_index_imfs2[k]) + '_' + str(time_min) + '&'+
  str(delay_min) + '.png')
plt.close(fig)

params.append([[time_min, delay_min, num_imfs1 - start_index_imfs1[k],
  num_imfs2 - start_index_imfs2[k]], max_vals, xcorr_map, xcorr_lags])
return params

#Numba function test: needed for it to perform better
x1 = psp_data[1]
x2 = solo_data[1]
test = cross_corr(x1, x2)
#-----
#Main part of the program: It performs the cross-correlation scan
#on the time windows and on the number of IMFS for some time windows.
#-----
time_mins = np.arange(10, 90, 2)
delay_min = 2
params_time = []

for i in time_mins:
  params_time.append(params_xcorr(1, i, delay_min))
#=====

```

---

## Referencias bibliográficas

---

- [1] M. G. Umran S. Inan, *Principles of Plasma Physics for Engineers and Scientists*, en (Cambridge University Press, 2011).
- [2] F. F. Chen, *Introduction to Plasma Physics and Controlled Fusion*, en (Springer International Publishing, Cham, 2016).
- [3] G. Parks, *Physics of space plasmas: an introduction*, en, 2nd edition (Westview Press, 2003).
- [4] A. W. Krall N. A.; Trivelpiece, *Principles of Plasma Physics* (1973).
- [5] T. E. Cravens, *Physics of solar system plasmas*, Cambridge Atmospheric and Space Science Series (Cambridge University Press, 1997).
- [6] Y. Kamide y A. C.-L. Chian, eds., *Handbook of the solar-terrestrial environment*, en (Springer, Berlin ; New York, 2007).
- [7] T. Encrenaz, J.-P. Bibring, M. Blanc, M.-A. Barucci, F. Roques y P. Zarka, *The Solar System*, en, ed. por I. Appenzeller, G. Börner, M. A. Dopita, M. Harwit, R. Kippenhahn, J. Lequeux, A. Maeder y V. Trimble, Astronomy and Astrophysics Library (Springer Berlin Heidelberg, Berlin, Heidelberg, 2004).
- [8] N. J. Fox, M. C. Velli, S. D. Bale, R. Decker, A. Driesman, R. A. Howard, J. C. Kasper, J. Kinnison, M. Kusterer, D. Lario, M. K. Lockwood, D. J. McComas, N. E. Raouafi y A. Szabo, en, [Space Science Reviews](#) **204**, 7 (2016).
- [9] D. Müller, O. C. St. Cyr, I. Zouganelis, H. R. Gilbert, R. Marsden, T. Nieves-Chinchilla, E. Antonucci, F. Auchère, D. Berghmans, T. S. Horbury, R. A. Howard, S. Krucker, M. Maksimovic, C. J. Owen, P. Rochus, J. Rodriguez-Pacheco, M. Romoli, S. K. Solanki, R. Bruno, M. Carlsson, A. Fludra, L. Harra, D. M. Hassler, S. Livi, P. Louarn, H. Peter, U. Schühle, L. Teriaca, J. C. del Toro Iniesta, R. F. Wimmer-Schweingruber, E. Marsch, M. Velli, A. De Groof, A. Walsh y D. Williams, en, [Astronomy & Astrophysics](#) **642**, A1 (2020).

- [10] M. Velli, L. K. Harra, A. Vourlidas, N. Schwadron, O. Panasenco, P. C. Liewer, D. Müller, I. Zouganelis, O. C. St Cyr, H. Gilbert, T. Nieves-Chinchilla, F. Auchère, D. Berghmans, A. Fludra, T. S. Horbury, R. A. Howard, S. Krucker, M. Maksimovic, C. J. Owen, J. Rodríguez-Pacheco, M. Romoli, S. K. Solanki, R. F. Wimmer-Schweingruber, S. Bale, J. Kasper, D. J. McComas, N. Raouafi, V. Martinez-Pillet, A. P. Walsh, A. De Groof y D. Williams, en, [Astronomy & Astrophysics](#) **642**, A4 (2020).
- [11] M. B. Priestley, *Non-linear and Non-stationary Time Series Analysis* (Academic Press, 1988).
- [12] C. Chatfield y H. Xing, *The analysis of time series: an introduction with R*, en, Seventh edition, Chapman & Hall/CRC texts in statistical science series (CRC Press, Taylor & Francis Group, Boca Raton, 2019).
- [13] R. Bruno y V. Carbone, *Turbulence in the Solar Wind*, en, vol. 928, Lecture Notes in Physics (Springer International Publishing, Cham, 2016).
- [14] W. H. Matthaeus, M. Wan, S. Servidio, A. Greco, K. T. Osman, S. Oughton y P. Dmitruk, en, [Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences](#) **373**, 20140154 (2015).
- [15] T. Alberti, G. Consolini, V. Carbone, E. Yordanova, M. Marcucci y P. De Michelis, [Entropy](#) **21**, 320 (2019).
- [16] E. M. L. L. D. Landau, *Fluid Mechanics*, 2.<sup>a</sup> ed. (Elsevier, 1987).
- [17] J. M. C. Yunus Çengel, *Fluid Mechanics Fundamentals and Applications*, 4.<sup>a</sup> ed. (McGraw-Hill College, 2017).
- [18] P. A. Davidson, *Turbulence: an introduction for scientists and engineers*, en, Second edition, OCLC: ocn911091654 (Oxford University Press, Oxford, United Kingdom ; New York, NY, United States of America, 2015).
- [19] L. Sorriso-Valvo, V. Carbone, P. Veltri, G. Consolini y R. Bruno, en, [Geophysical Research Letters](#) **26**, 1801 (1999).
- [20] B. Hnat, S. C. Chapman y G. Rowlands, en, [Physical Review E](#) **67**, 056404 (2003).
- [21] P. J. Coleman Jr., en, [The Astrophysical Journal](#) **153**, 371 (1968).
- [22] R. Bruno y V. Carbone, en, [Living Reviews in Solar Physics](#) **10**, 10.12942/lrsp-2013-2 (2013).
- [23] R. D'Amicis, R. Bruno, G. Pallochia, B. Bavassano, D. Telloni, V. Carbone y A. Balogh, en, [The Astrophysical Journal](#) **717**, 474 (2010).
- [24] R. Bruno, L. Trenchi y D. Telloni, en, [The Astrophysical Journal](#) **793**, arXiv:1408.4595 [astro-ph, physics:physics], L15 (2014).

- [25] R. Bruno y L. Trenchi, en, [The Astrophysical Journal](#) **787**, L24 (2014).
- [26] S. J. Schwartz y E. Marsch, en, [Journal of Geophysical Research](#) **88**, 9919 (1983).
- [27] G. P. Zank, L. Adhikari, P. Hunana, D. Shiota, R. Bruno y D. Telloni, en, [The Astrophysical Journal](#) **835**, 147 (2017).
- [28] D. Telloni, L. Sorriso-Valvo, L. D. Woodham, O. Panasenco, M. Velli, F. Carbone, G. P. Zank, R. Bruno, D. Perrone, M. Nakanotani, C. Shi, R. D'Amicis, R. De Marco, V. K. Jagarlamudi, K. Steinvall, R. Marino, L. Adhikari, L. Zhao, H. Liang, A. Tenerani, R. Laker, T. S. Horbury, S. D. Bale, M. Pulupa, D. M. Malaspina, R. J. MacDowall, K. Goetz, T. D. de Wit, P. R. Harvey, J. C. Kasper, K. E. Korreck, D. Larson, A. W. Case, M. L. Stevens, P. Whittlesey, R. Livi, C. J. Owen, S. Livi, P. Louarn, E. Antonucci, M. Romoli, H. O'Brien, V. Evans y V. Angelini, en, [The Astrophysical Journal Letters](#) **912**, L21 (2021).
- [29] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung y H. H. Liu, en, [Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences](#) **454**, 903 (1998).
- [30] A. T. W. Donald B. Percival, *Wavelet Methods for Time Series Analysis* (Cambridge University Press, 2013).
- [31] P. De Michelis, G. Consolini y R. Tozzi, en, [Nonlinear Processes in Geophysics](#) **19**, 667 (2012).
- [32] F. Carbone, L. Sorriso-Valvo, T. Alberti, F. Lepreti, C. H. K. Chen, Z. Němeček y J. Šafránková, en, [The Astrophysical Journal](#) **859**, 27 (2018).
- [33] F. Carbone, D. Telloni, L. Sorriso-Valvo, G. Zank, L. Zhao, L. Adhikari y R. Bruno, en, [Universe](#) **6**, 116 (2020).
- [34] F. Carbone, L. Sorriso-Valvo, Y. V. Khotyaintsev, K. Steinvall, A. Vecchio, D. Telloni, E. Yordanova, D. B. Graham, N. J. T. Edberg, A. I. Eriksson, E. P. G. Johansson, C. L. Vásconez, M. Maksimovic, R. Bruno, R. D'Amicis, S. D. Bale, T. Chust, V. Krasnoselskikh, M. Kretzschmar, E. Lorfèvre, D. Plettemeier, J. Souček, M. Steller, Š. Štverák, P. Trávníček, A. Vaivads, T. S. Horbury, H. O'Brien, V. Angelini y V. Evans, en, [Astronomy & Astrophysics](#) **656**, A16 (2021).
- [35] S. H. Strogatz, *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*, en, Second edition, OCLC: ocn842877119 (Westview Press, a member of the Perseus Books Group, Boulder, CO, 2015).
- [36] K. Rao, D. Kim y J.-J. Hwang, *Fast Fourier Transform - Algorithms and Applications*, en, Signals and Communication Technology (Springer Netherlands, Dordrecht, 2010).

- [37] Y. X. Huang, F. G. Schmitt, Z. M. Lu e Y. L. Liu, en, [EPL \(Europhysics Letters\) \*\*84\*\*, arXiv:1401.4211 \[physics\], 40010 \(2008\)](#).
- [38] G. Rilling, P. Flandrin y P. Goncalves, en, 5 (2003).
- [39] P. Flandrin, G. Rilling y P. Goncalves, en, [IEEE Signal Processing Letters \*\*11\*\*, 112 \(2004\)](#).
- [40] T. Nguyen y G. Strang, *Wavelets and Filter Banks*, 2nd (Wellesley College, 1996).
- [41] T. Pang y N. EBSCO Publishing (Firm) Inc, *An introduction to computational physics*, en, OCLC: 1236196394 (Cambridge University Press, Cambridge, 2006).
- [42] T. Young y M. J. Mohlenkamp, *Introduction to Numerical Methods and Matlab Programming for Engineers* (2021).
- [43] A. Quinn, V. Lopes-dos-Santos, D. Dupret, A. Nobre y M. Woolrich, en, [Journal of Open Source Software \*\*6\*\*, 2977 \(2021\)](#).
- [44] D. Duan, J. He, T. A. Bowen, L. D. Woodham, T. Wang, C. H. K. Chen, A. Mallet y S. D. Bale, en, [The Astrophysical Journal Letters \*\*915\*\*, arXiv:2102.13294 \[astro-ph, physics:physics\], L8 \(2021\)](#).
- [45] R. Bruno, B. Bavassano, E. Pietropaolo, V. Carbone y P. Veltri, en, [Geophysical Research Letters \*\*26\*\*, 3185 \(1999\)](#).
- [46] J. W. Belcher y L. Davis, en, [Journal of Geophysical Research \*\*76\*\*, 3534 \(1971\)](#).