

Sobre Expresiones Cerradas para Ondas Gravitacionales

por

Aarón Villanueva

Presentado ante la Facultad de Matemática, Astronomía, Física y Computación como parte de los requerimientos para la obtención del grado de

Licenciado en Física

de la

UNIVERSIDAD NACIONAL DE CÓRDOBA



©FAMAF – UNC 2019

Director: Dr. Manuel Tiglio

Esta obra está bajo una licencia Creative Commons “Reconocimiento-NoCommercial 4.0 Internacional”.



Agradecimientos

Agradezco a mi director Manuel Tiglio por el constante apoyo para llevar a cabo este proyecto e introducirme en este excitante campo de investigación. A mi familia y amigos por el apoyo moral a lo largo de la carrera. A mi pareja Alejandra por la compañía incansable en la elaboración y redacción de este trabajo.

Este proyecto fue financiado en parte por subsidio NSF1607646 a la Universidad de California San Diego, e inicialmente ideado por Rory Smith y Manuel Tiglio, quien agradece la hospitalidad de Alan Weinstein, Yanbei Chen y Kip Thorne en CalTech durante ese período. Agradecimientos a Hod Lipson (Cornell, Columbia University) por introducirnos en programación genética y al equipo de Nutonian por facilitarnos el uso gratuito de Eureka.

Resumen

Uno de los desafíos más acuciantes que surgen en Ciencia de Datos es el manejo de sistemas complejos (incluyendo análisis y construcción de modelos predictivos) y de grandes cantidades de información (*Big Data*). Una característica común de estos escenarios es el costo computacional involucrado, que en muchos casos puede llegar a ser prohibitivo, al punto de no poder resolverse con el uso de supercomputadoras, hardware especializado ni optimización de software. Esto se exagera en el caso de problemas parametrizados, ya que el costo computacional puede crecer rápidamente, a veces de forma exponencial, con el número de parámetros que caracterizan el *input* del problema, con riesgo de caer así en la llamada 'maldición de la dimensionalidad' (*curse of dimensionality*). Un enfoque para solucionar dicho problema es disminuir la fidelidad, i.e. reemplazar el modelo fundamental (cuando existe y es conocido) por un modelo aproximado, computacionalmente más económico. Esto es delicado y poco confiable ya que, en principio, puede desconocerse la precisión de la aproximación si no se tiene acceso al modelo exacto (o a uno de alta fidelidad); esto es, una falta de acceso a *truth/ground solutions* para validación. En esta tesis presentamos un enfoque novedoso para encontrar expresiones cerradas de alta fidelidad, **ab-initio** -esto es, sin simplificar el modelo subyacente u original, en caso de existir uno- en el campo de predicción y análisis de datos de sistemas complejos, evitando así incurrir en la maldición de la dimensionalidad. Este enfoque es *data-driven*, de modo que no es necesario tener acceso a un modelo fundamental.

En este trabajo implementamos, en sinergia, Bases Reducidas, el Método de Interpolación Empírica -en teoría de aproximaciones, se consideran ambos métodos casi-óptimos en un sentido matemático riguroso- y Programación Genética para Regresión Simbólica. La aplicación presentada corresponde al ámbito de ondas gravitacionales emitidas en la colisión de dos agujeros negros, restringido al caso sin espín. En lugar de incurrir en meses de supercomputación de relatividad numérica para resolver las ecuaciones de Einstein en cada caso de interés, presentamos expresiones cerradas de alta precisión y de un par de líneas, que se pueden evaluar rápidamente en un computador personal. Así, realizamos una demostración de principio altamente no trivial para ondas gravitacionales. Este enfoque debería tener múltiples aplicaciones en diferentes disciplinas de Ciencia de Datos. Algunas de sus ventajas son, por un lado, que implica un sistema de aprendizaje altamente paralelizable y, por otro, que convierte un problema de aprendizaje de complejidad combinatorial en uno de complejidad constante, jerárquico y agnóstico (*data-driven*).

Abstract

One of the most pressing challenges appearing in Data Science is the management of complex systems (including analysis and construction of predictive models) and dealing with big amounts of information (Big Data). A common characteristic of these scenarios is the computational cost involved that, in most cases, can be prohibitively expensive, to the point of not being able to resolve with the assistance of supercomputers, specialized hardware or software optimization. This is exacerbated in the case of parameterized systems, since the computational cost can grow rapidly -sometimes exponentially- with the number characterizing the input of the problem, usually resulting in the so called 'curse of dimensionality'. An approach to face this problem is decreasing the fidelity, i.e. replacing the fundamental model (when this exists and it is known) by an approximate model, computationally cheaper. This is delicate and unreliable because, in principle, the precision of the approximation can be unknown if one hasn't have access to an exact model (or to a high fidelity one); that is, lack of access to truth/ground solutions for validation. We present in this thesis a new approach to find high-fidelity closed forms, **ab-initio** -that is, without simplify the underlying model, in case of existence- in the field of prediction and data analysis of complex systems, avoiding the curse of dimensionality. This approach is data-driven, so there is no necessity of invoking an underlying model.

In this work, we implement, in synergy, Reduced Basis, the Empirical Interpolation Method -in approximation theory, both are considered quasi-optimal in a rigorous mathematical sense- and Genetic Programming for Symbolic Regression. The application corresponds to gravitational waves produced by a non spinning binary black hole coalescence. In place of months of numerical relativity simulations with supercomputers in order to solve Einstein's equations for each case, we present high precision closed forms of a couple of lines, that can be rapidly evaluated in a personal computer. Therefore, we carry out a highly non trivial demonstration of principle for gravitational waves. This approach would have multiple applications in different branches of Data Science. Some of its advantages are, on the one hand, that implies a system of high parallelizability and, by the other, that turns a learning problem of combinatorical complexity to one of constant, hierarchical and agnostic (data-driven) complexity.

Índice general

1. Introducción y Visión General	5
2. Enfoque de esta Tesis	9
3. Resultados: un panorama	11
3.1. Introducción	11
3.2. Regresión Simbólica para una sola onda	14
3.3. Regresión Simbólica para todas las ondas	17
3.3.1. Descomposición de dominio	20
4. Teoría de Aproximaciones: Aprendizaje y Bases Reducidas	23
4.1. Aproximación por proyección	23
4.1.1. Errores colectivos	24
4.1.2. Medida n-Kolmogorov	24
4.2. Bases Reducidas	26
4.2.1. Algoritmo Greedy	27
4.2.2. Convergencia del método greedy	28
5. Modelos Predictivos	31
5.1. De Proyección a Interpolación	31
5.2. Método de Interpolación Empírica (EIM)	33
5.2.1. Algoritmo EIM	33
5.2.2. Condicionamiento y Convergencia del EIM	34
5.3. Modelos sustitutos	37
6. Programación Genética	39
6.1. Introducción	39
6.2. Regresión Simbólica (RS)	40
6.2.1. RS para encontrar leyes de la naturaleza	42
7. Resultados generales	45
7.1. Buscando formas cerradas	45
7.2. Los nodos EIM en acción	46
7.3. Resultados	47
7.4. Validación	49
8. Conclusión	52
A. Errores en la fase de ondas gravitacionales	54

Capítulo 1

Introducción y Visión General

Debido al incremento masivo de información digital en los diferentes campos de la actividad humana (ciencias duras, cognitivas, inteligencia de negocios, *analytics* en general, biología, economía, medicina, ciencias sociales, etc.), la necesidad de poder analizar, procesar, sintetizar y hallar patrones ocultos de forma eficiente, generando modelos a partir de su estudio, se ha convertido en algo imprescindible al momento de aprovechar el campo de información accesible. La ciencia aplicada al análisis de datos y, en particular, a grandes cantidades del mismo (*Big Data*), usando técnicas modernas de aprendizaje computacional, acuña el nombre de Ciencia de Datos (*Data Science*). Cuando en un campo de estudio particular la cantidad de datos crece, se vuelve necesario poder estudiar las posibles redundancias presentes en el conjunto y encontrar relaciones ocultas para *comprender* globalmente el mismo y *expresarlo* de la manera más sucinta posible. Esto nos conduce naturalmente al aprovechamiento de dicha *reducción* para poder generar información nueva, es decir, avanzar a la instancia predictiva del problema de interés para ganar información. Aparece entonces en Ciencia de Datos el beneficio de la construcción de modelos predictivos y herramientas de análisis de datos mediante la posible aplicación de aprendizaje no supervisado (Inteligencia Artificial y Machine Learning), a partir de la constitución de modelos reducidos compactos, de resolución arbitrariamente alta y rápidos de evaluar.

En Relatividad General las ecuaciones de Einstein (EE) son un conjunto de 10 ecuaciones diferenciales elíptico-hiperbólicas, acopladas para las componentes del tensor métrico que describe la geometría del espaciotiempo. Son unas de las ecuaciones en derivadas parciales más complejas y computacionalmente intensivas de resolver, dentro de aquellas que describen leyes fundamentales de la naturaleza. La implementación de métodos analíticos para obtener soluciones exactas ha estado limitada a modelos con simetrías que, si bien arrojan información conceptual, no son casos de interés en la práctica (e.g., en ciencia de ondas gravitacionales). Por otro lado, métodos numéricos que resuelvan con suficiente resolución escenarios de interés observacional han sido aplicados con éxito sólo en la última década [5, 9, 31], después de varios años de un gran esfuerzo colectivo, aunque a un alto costo (super)computacional *por configuración*.

Uno de estos escenarios es la coalescencia de objetos compactos como agujeros negros, estrellas de neutrones o procesos mixtos. Las ecuaciones de Einstein predicen la generación de ondas en la estructura del espaciotiempo en dichos procesos. Debido a la baja relación señal/ruido (SNR, *signal to noise ratio*) que proveen interferómetros de ondas gravitacionales tales como LIGO o VIRGO, para detectar estas ondas, se necesita comparar las señales con soluciones numéricas de las EE y así poder afirmar si la

señal corresponde efectivamente a una onda gravitacional o no. A este efecto, se debe contar con un banco de modelos que cubran densamente el espacio de parámetros de interés. Posterior a una detección, el siguiente paso es realizar inferencia de parámetros que incluyen: 1) parámetros intrínsecos: aquellos de las fuentes tales como las masas y espines de cada agujero negro y 2) parámetros extrínsecos: localización en el espacio (tales como distancia a la tierra), ángulo de incidencia y tiempo de llegada de la señal.

La relevancia de esta tesis

Una sola simulación de dos agujeros negros lleva alrededor de 10,000 horas reloj *mínimo*. A esta cantidad contribuyen los múltiples procesos de corrección y validación que orbitan alrededor de la solución cruda, tales como tests de convergencia, la realización pruebas para reducir la excentricidad del dato inicial, etc., sumando a esto las contingencias que induzcan a repetir las simulaciones. La cantidad exacta de horas depende de factores como el código y la precisión del mismo. 10,000 horas por configuración es un número optimista que nadie en la comunidad de relatividad numérica objetaría.

Supongamos entonces que se samplean las 8 dimensiones (una masa y tres componentes de espín por objeto) de dos agujeros negros con 100 puntos por dimensión, ya sea uniforme o aleatoriamente. El tiempo computacional para todas esas simulaciones sería de 10^{16} años (órdenes de magnitud más que la edad del universo, que es alrededor de 13,8 billones de años). Se podría decir, en respuesta a este argumento, que para eso existen las supercomputadoras. Bien, asumamos que se tiene acceso continuo a las 10 supercomputadoras más poderosas del mundo, trabajando todas ellas en paralelo. Eso es equivalente a tener 22.419.648 procesadores¹. Las chances de que aquello sea factible pueden ser altamente debatibles, pero sigamos con el argumento: con esos recursos computacionales, el tiempo de cómputo disminuiría a aproximadamente $5 \cdot 10^8$ años.

A principios del 2000, en el auge de *grid computing*, se hablaba de 'conectar las supercomputadoras más grandes del mundo para resolver este problema de forma épica'. No se previó que esta idea no era una solución, aún sin considerar *latencia*, por lo que decayó eventualmente hasta desaparecer para esta aplicación².

Desde los '90 el objetivo de observatorios como LIGO fue testear la teoría de Relatividad General en su régimen más extremo y no lineal y detectar por primera vez, de forma directa, ondas gravitacionales, usando simulaciones numéricas de las EE como catálogos para análisis de datos. Curiosamente, este problema de 'sampleo' pasó desapercibido, ignorado, relegado, o una combinación de todo ello, en la comunidad de ondas gravitacionales, hasta que las primeras simulaciones de agujeros negros estables fueron posibles a mediados de los 2000.

Una vez comprendido el problema del costo computacional, comenzaron a desarrollarse modelos aproximados con errores desconocidos, que no vamos a repasar porque el objetivo de este trabajo es dejarlos de lado y trabajar únicamente con las EE, sin ninguna aproximación física (enfoque *ab-initio* o *de primeros principios*).

De forma subsecuente, hacer inferencia de parámetros de la fuente de la señal detectada hereda un problema similar, dado que para llevar a cabo inferencia Bayesiana (enfoque unánimemente consensuado como más apropiado que un análisis frecuentista) se necesitan computar en tiempo real ondas gravitacionales *on demand*. Esto es: se desconocen los parámetros que se requieren hasta analizar un parámetro tras otro, de forma secuencial. Sin la introducción de técnicas modernas de aprendizaje, este proce-

¹<https://www.top500.org/lists/2019/06/>

²*grid computing*, en la actualidad, tiene muchas otras aplicaciones activas y de alto valor.

so hubiera llevado meses de análisis por señal [4, 10, 11] cuando, en la actualidad, la frecuencia de detecciones de ondas gravitacionales está creciendo. Por ejemplo, en los últimos dos años, LIGO ha anunciado la detección de 11 eventos (10 correspondientes a la colisión de dos agujeros negros y 1 a la colisión binaria de dos estrellas de neutrones).

El panorama en la práctica no ha sido tan oscuro como la exposición previa puede sugerir: lo que se hace es usar modelos aproximados para la búsqueda de ondas y, si hay una detección, se usa una cantidad ingente de técnicas aproximadas para estimar los parámetros de la fuente y así contrastar los resultados de distintas aproximaciones. Una vez que una región del espacio de parámetros suficientemente pequeña es confiada, se realizan simulaciones de supercomputadoras en dicha región. Este es un proceso no solamente tedioso, sino ineficiente en términos de precisión, recursos humanos y computacionales.

En resumen, el problema para ondas gravitacionales es evidente: reducir el costo computacional a un tiempo real o cuasireal para poder predecir ondas gravitacionales de forma fidedigna. Este problema fue resuelto para expresiones post-newtonianas [20] [13], para el *ringdown* de un agujero negro, usando varios modos para testear el teorema del *no-hair* [13], para la colisión de dos agujeros negros inicialmente sin espín [8], e incluso para el diseño de interferometría de alta precisión [15], usando un paradigma disruptivo. El formalismo transversal a todas estas aplicaciones está resumido en [21] y se describe en esta tesis. El caso genérico multidimensional, si bien es un área muy activa y con rápido progreso en teoría de aproximaciones y aprendizaje, sigue siendo un problema esencialmente abierto³.

En esta tesis tomamos el problema de la colisión de dos agujeros negros inicialmente sin espín, para el cual se construyeron modelos *surrogates* que son completamente equivalentes a soluciones de supercomputadoras, pero se pueden evaluar en un notebook en alrededor de 0.01 segundo por modo multipolar [8]. La innovación en esta tesis es el uso de estos modelos reducidos y subsamples adaptativos en el tiempo, con el fin de construir expresiones *cerradas* para ondas gravitacionales, usando Regresión Simbólica. Como prueba de concepto, nos enfocamos en el modo multipolar ($\ell = 2, m = 2$), dominante en colisiones de agujeros negros, inicialmente en órbita cuasi-circular (el caso de mayor interés para LIGO-VIRGO). Cualquier otro modo sigue un desarrollo similar que se publicará en el futuro o será dejado a la comunidad para su implementación: siguiendo los pasos aquí descritos, esto representa un procedimiento conceptualmente directo (aunque no simple en la práctica).

En un sentido abstracto, por diseño, esta tesis aporta a teoría de aproximaciones, bases reducidas, grillas *sparse* y programación genética, con componentes de computación científica, análisis numérico y modelado matemático para problemas parametrizados. La aplicación aquí presentada es sobre ondas gravitacionales, pero los desarrollos son genéricos y deberían ser aplicables a otras disciplinas en Ciencia de Datos.

³Un review sobre Ciencia de Datos en Relatividad General es [24]

Capítulo 2

Enfoque de esta Tesis

La teoría y metodología usadas en esta tesis están presentadas en detalle técnico en los capítulos 4, 5, 6 y 7. Aquí simplemente haremos un bosquejo conceptual del procedimiento. Los pasos están ordenados de la siguiente manera:

1. Construir una representación casi-óptima del sistema parametrizado de interés (ondas gravitacionales, en el caso de la aplicación aquí presentada), aprovechando cualquier regularidad existente respecto de variación en parámetros. Esto involucra construir un conjunto *sparse* en el espacio de parámetros y una representación compacta y de alta resolución usando Bases Reducidas [7] - enfoque de elección en la actualidad para modelos reducidos (ROM, por sus siglas en inglés, *Reduced Order Modeling*)-.
2. Construir un interpolante empírico usando el Empirical Interpolation Method (EIM)[6, 14], también casi-óptimo, como primer paso para desarrollar un modelo predictivo *en parámetros y tiempo* [21]. El EIM involucra construir un conjunto *sparse* de nodos *temporales*, dual al de parámetros¹, y un interpolante con una precisión aproximada a la de la base reducida, reemplazando proyección por interpolación. De esta forma se restringe aún más la información requerida para el siguiente paso.
3. Construir el modelo predictivo (MP) para el caso de aplicación de esta tesis: las ondas gravitacionales emitidas en la colisión binaria de dos agujeros negros. Para esto se usan los resultados de [8]. Este MP es *numérico*.
4. Usar herramientas modernas de Regresión Simbólica [35, 36] y Programación Genética para construir formas cerradas (de aquí en adelante, *modelos simbólicos*) representando con alta precisión el MP numérico y, por ende, simulaciones de supercomputadoras de las ecuaciones de Einstein.
5. Testear y validar el modelo simbólico respecto al MP numérico. Dado que los MP numéricos han sido testeados y validados respecto de soluciones de supercomputadoras de relatividad numérica de alta precisión y que han demostrado ser tan fidedignos como las últimas [8, 21, 22], testear y validar las expresiones simbólicas es equivalente a hacerlo respecto de simulaciones de supercomputadoras ².

¹Este enfoque es más eficiente que el sampleo de Nyquist [37], una técnica histórica muy usada en procesamiento de señales.

²Cada uno de los pasos (1-4) involucra entrenamiento no supervisado y validación. Algunos de ellos se explican en este trabajo; otros, por brevedad, se referencian a literatura existente y en preparación.

El resultado principal de esta tesis es que, siguiendo el procedimiento recién descrito, se pueden reemplazar, con alta resolución, simulaciones en supercomputadoras de las EE por un par de líneas de expresiones cerradas

Capítulo 3

Resultados: un panorama

3.1. Introducción

Presentamos un panorama general de algunos de los resultados obtenidos¹.

Una onda gravitacional genérica tiene dos componentes dependientes del tiempo,

$$h_+(t, \boldsymbol{\lambda}) \quad y \quad h_x(t, \boldsymbol{\lambda}),$$

que representan los dos grados de polarización de la teoría (como en electromagnetismo, son ondas transversales); el índice $\boldsymbol{\lambda}$ (que guarda información de las masas, espines, etc., del sistema que generó las ondas) denota los diferentes perfiles posibles en el espacio de parámetros. En esta tesis la dimensión paramétrica es unidimensional, correspondiente al cociente entre las masas $q := m_1/m_2$, i.e., $\boldsymbol{\lambda} = q$. Como prueba de concepto, nos centramos en el modo multipolar ($\ell = 2, m = 2$), que es el modo dominante en la colisión de dos agujeros negros en órbita inicial cuasi-circular.

Al variar q se obtienen diferentes perfiles de ondas gravitacionales correspondientes a agujeros negros de distintos cocientes de masa. En Relatividad General, la masa total es una escala global, de modo que conociendo sólo la onda para q se pueden construir ondas para cualquier par de masas².

Siguiendo la notación estándar en la literatura, las dos componentes mencionadas se unen en una sola función compleja

$$h(t, q) := h_+(t, q) + i h_x(t, q), \tag{3.1}$$

donde i denota la unidad imaginaria. Esto, como veremos en breve, más allá de simple notación, resulta conveniente en la práctica. El perfil de estas ondas es oscilatorio, con una amplitud creciente hasta el momento de la coalescencia de los agujeros negros, seguido por un *ringdown* convergente a un agujero negro estacionario³.

El régimen inicial de la coalescencia de dos objetos masivos orbitando uno alrededor del otro antes de la fusión se denomina *inspiral*; el período corto que corresponde a la coalescencia se denomina *merger* y corresponde al pico de la amplitud de las ondas; finalmente, en la fusión, ocurre un período de oscilación amortiguada denominado *ringdown*. En la figura 3.1 pueden verse las tres instancias representadas como la parte

¹Para resultados generales ver el capítulo 7.

²Para una introducción clara y concisa a la ciencia de ondas gravitacionales, véase [27, 28].

³Esta componente, para la audiencia de ondas gravitacionales, recae en la estabilidad de Kerr (que no ha sido completamente demostrada)

larga a la izquierda, el pico al medio, y la parte decreciente a la derecha, respectivamente.

Valores de referencia (ground truth)

En los capítulos 4 y 5 describiremos cómo construir modelos sustitutos (*surrogates*) que son rápidos de evaluar (alrededor de 0.01 s por modo multipolar en un computador portátil) y completamente equivalentes a simulaciones numéricas de las EE usando supercomputadoras. Implementaremos estos modelos sustitutos para generar ondas **de referencia** (*ground truth*) usando el paquete *gwsurrogate* [2]. Este paquete es un código abierto escrito en lenguaje *python* que implementa ROM+EIM+MP⁴, resumidos en modelos sustitutos numéricos para ondas gravitacionales, sin pérdida de resolución respecto de relatividad numérica, pero de rápida evaluación.

El modelo sustituto usado en esta tesis fue entrenado implementando simulaciones de relatividad numérica en el rango $q \in [1, 10]$ para agujeros negros inicialmente en órbita cuasi-circular y sin espín, por alrededor de 15 ciclos para cada onda. Más precisamente, en el rango $t \in [-2750, 100]M$, donde M es la masa total (ADM) del espaciotiempo. Siguiendo la convención en ciencia de ondas gravitacionales, las ondas se alinean de forma tal que $t = 0$ corresponda al pico de la amplitud (el *merger*).

Velocidad de convergencia en Regresión Simbólica

Cuando hablamos de convergencia rápida en regresión simbólica, nos referimos a minutos en un computador portátil u ordenador estándar; cuando hablamos de lo contrario, nos referimos a más de una semana en un ordenador con varios núcleos, sin progreso alguno en la convergencia.

Medidas de errores

Usamos dos nociones de errores aplicadas a los modelos simbólicos y sus contrapartes de referencia:

- Para amplitud y fase usamos el coeficiente de determinación (R^2 , *goodness of fit*), definido como

$$R^2 := 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}},$$

donde SS_{res} es la suma de los cuadrados residuales entre valores simbólicos y aquellos de referencia y SS_{tot} es la suma de los cuadrados de las diferencias entre valores de referencia y su valor medio. El rango R^2 es $[0, 1]$; cuanto más alto es el valor, ‘mejor’ es la capacidad predictiva del modelo.

La razón para no usar errores relativos para amplitud y fase reside en que estos se aproximan a cero para ciertos valores de tiempo, produciendo errores de redondeo.

- Para las ondas usamos el ‘overlap’ S , definido como

$$S(h_o, h_s)(q) = \frac{|\langle h_o(q) | h_s(q) \rangle|}{\|h_o(q)\| \|h_s(q)\|}, \quad (3.2)$$

donde h_o :=onda de referencia y h_s := onda simbólica. Por la desigualdad de Cauchy-Schwarz, $0 \leq S \leq 1$. El caso $S = 1$ corresponde a dos ondas ‘coinci-

⁴Reduced Order Modeling+Empirical Interpolation Method+Modelo Predictivo

dentes' ⁵. Este estimador es usado en análisis de datos, en particular en ciencia de ondas gravitacionales, tanto para detección como para estimación de parámetros.

Validación del modelo simbólico

Usamos varios criterios para validar las expresiones simbólicas presentadas:

1. Dividir los datos numéricos en dos conjuntos: uno para entrenamiento y otro para validación. Típicamente usamos 50% de los datos para cada uno de ellos y, en cada iteración de la búsqueda, se reelige de forma aleatoria el 50% a usar para cada parte. Esto se hace tanto para la amplitud como para la fase.
2. Una vez elegidos modelos simbólicos para amplitud y fase, se los compara con nuevas funciones de referencia en un conjunto denso que *no* hayan estado en el conjunto inicial descrito en (1). Esto es posible porque el modelo sustituto se puede evaluar de forma muy rápida.
3. El proceso (2) se repite para ondas reconstruidas a partir de amplitud y fase simbólicas, comparándolas con sus contrapartes de referencia. Nuevamente, esto es posible por la velocidad de evaluación del modelo sustituto.

En resumen, el proceso de validación es bastante completo y, en particular, asegura que no se cometa *over-fitting*, otorgando alta confianza a los resultados obtenidos.

Descomposición en amplitud y fase

Si desplegamos cada superficie $h_+(t, q)$, $h_x(t, q)$ en \mathbb{R}^3 , la fig. 3.1 (donde las curvas oscilatorias representan un corte para un q específico) nos sugiere de antemano que la complejidad de dichas superficies debe ser alta, pues cada onda varía en modulación y frecuencia en todo el dominio (t, q) . No obstante, al descomponer la onda $h(t, q)$ de la ec. (3.1) en amplitud y fase,

$$h(t, q) = A(t, q) \times \exp \phi(t, q),$$

vemos que la complejidad aparente en los dos modos iniciales se simplifica en su representación polar. En la fig. 3.2 se observa la simpleza estructural de la amplitud y en la fig. 3.3 la de la fase.

La simplicidad de la estructura para la amplitud y la fase, en comparación con las ondas gravitacionales mismas, sugiere la estrategia a seguir en esta tesis: encontrar expresiones simbólicas para ellas y luego reconstruir los dos grados de polarización de las ondas:

$$h_+(t, q) = A(t, q) \cos(\phi(t, q)) \quad h_x(t, q) = A(t, q) \sin(\phi(t, q)). \quad (3.3)$$

En la siguiente sección damos nuestro primer ejemplo y resultados de este enfoque.

⁵En rigor, el caso $S = 1$ corresponde a $h_o = \alpha h_s$ para algún α real o complejo. Por lo que dos ondas proporcionales serán 'iguales' para el estimador S . Esto no es un problema y se tiene en cuenta en cada dominio de aplicación.

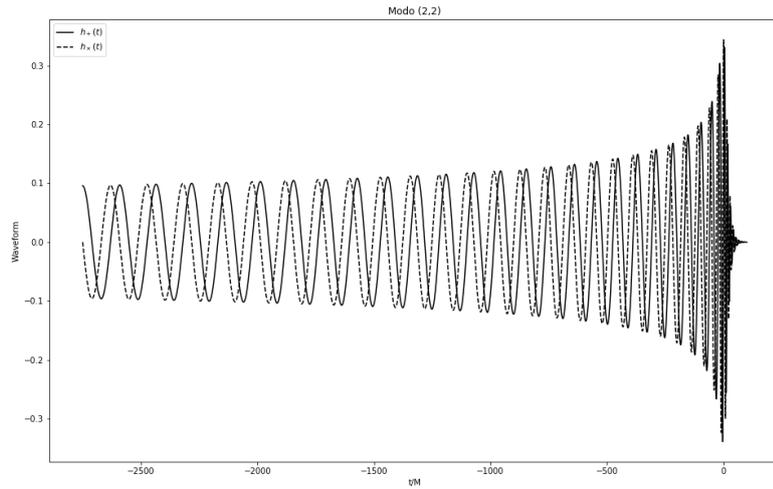


Figura 3.1: Los dos modos h_+ y h_x de polarización para $q = 2$ en función del tiempo.

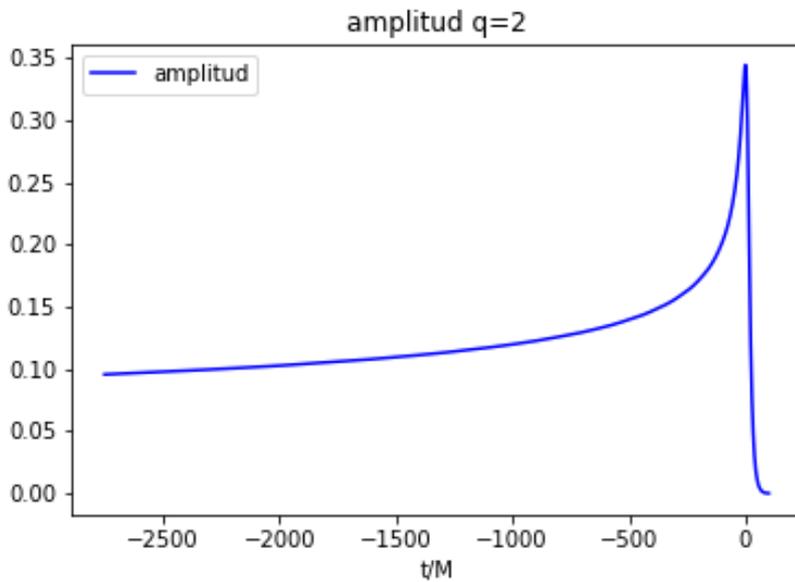


Figura 3.2: Amplitud $A(q = 2, t)$ para las ondas de la figura 3.1.

3.2. Regresión Simbólica para una sola onda

A modo de muestra, presentamos los resultados de RS para hallar formas cerradas para una única onda, i.e., correspondiente a un valor determinado del cociente entre las masas, $q = 2$ (esta elección fue arbitraria). Tanto amplitud como fase presentan sus propios desafíos al intentar una regresión simbólica, como describimos más adelante a través de las soluciones a los mismos.

Amplitud

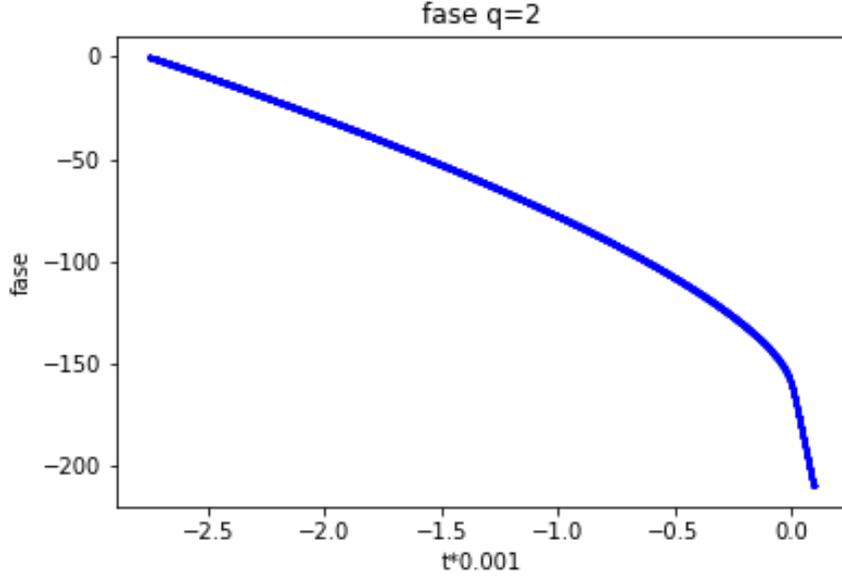


Figura 3.3: Fase $\phi(q = 2, t)$ para las ondas de la figura 3.1.

El desafío para la amplitud es resolver el pico sin usar una grilla temporal para entrenamiento tan densa que el algoritmo genético nunca converja (recordemos que la complejidad computacional de estos algoritmos, al menos en nuestra experiencia, crece rápidamente con la cantidad de datos -ver sección 6-). De hecho, probamos este camino sin obtener resultados positivos. La otra alternativa es usar una grilla con menos puntos temporales. Con esta estrategia, se obtienen modelos simbólicos con errores inaceptables, sobre todo en el pico y su región cercana, incluyendo el *ringdown*.

Esto motiva el uso de una grilla *sparse* en el tiempo, que automáticamente coloque puntos donde sea necesario, usando un criterio no supervisado y casi-óptimo. El Método de Interpolación Empírica, presentado en el cap. 5, reúne precisamente estas características y sirve de plataforma para construir un modelo predictivo.

Presentamos ahora los resultados obtenidos usando los valores de referencia de $A(q = 2, t)$ sólo en los nodos EIM temporales con el objeto de construir un conjunto de entrenamiento. Al ser una grilla tan *sparse*, el algoritmo genético converge rápidamente y representa la amplitud en todo el dominio temporal con alta precisión. La expresión cerrada resulta⁶

$$\begin{aligned}
A(t, q = 2) = & 0.0577749543380633 + 0.0153267125446358t + \\
& 0.321609005302022 \tanh(\tanh(t)) + 0.183858921753946 \\
& \operatorname{atan2}(0.0290636187282596 - t, -0.750887432348254t \\
& \operatorname{atan2}(-3.68070144826721t, 0.0272476233807906)) - \\
& 0.274421462807621 \tanh(t) \operatorname{atan2}(0.0290636187282596 - t, \\
& -0.750887432348254t \operatorname{atan2}(-3.68070144826721t, \\
& 0.0272476233807906))
\end{aligned} \tag{3.4}$$

⁶Por completitud, se muestran las constantes arrojadas por RS con todos los dígitos. A efectos prácticos, pueden haber dígitos espúreos que, en la valuación de la fidelidad del modelo simbólico, no afectan la resolución del mismo cuando es comparado con el de referencia.

El coeficiente de determinación, en todos los resultados presentados en esta tesis tanto para amplitud como fase, es de al menos $R^2 \sim 0.999$. La razón para no dar valores exactos es que lo realmente importante es la precisión en la reconstrucción de las ondas a partir de amplitud y fase, que de hecho es más demandante y presenta desafíos adicionales, como se explica más abajo.

Fase

La estructura de la fase es mucho más simple que la de la amplitud, como se puede apreciar en la figura 3.3. Por esa razón usamos para entrenamiento todos los puntos temporales producidos por *gwsurrogate* correspondientes a las ondas de referencia y, aún así, la regresión simbólica converge rápidamente. La expresión cerrada obtenida es

$$\begin{aligned} \phi(t, q = 2) = & - 161.956450430632 - 221.244404102037t \\ & - 0.457591076509569t^3 - 5.29027350365944t^2 - \\ & 10.0315028752836 \operatorname{asinh}(13.8643478324655t) - \\ & 167.072367427131t \operatorname{tanh}(20.2969567740124t) \end{aligned} \quad (3.5)$$

Reconstrucción de la onda a partir de amplitud y fase

La fig. 3.4 muestra la reconstrucción de los perfiles h_+ y h_x usando los modelos simbólicos para amplitud y fase versus las funciones de referencia. El overlap entre modelo simbólico y valores de referencia es de 0.979735 para h_+ y 0.979735 para h_x . **Que estos valores se correspondan no es coincidencia, dado que ambas polarizaciones solo difieren en un valor de fase $\pi/2$; por lo que solo hace falta testear/validar una de ellas.**

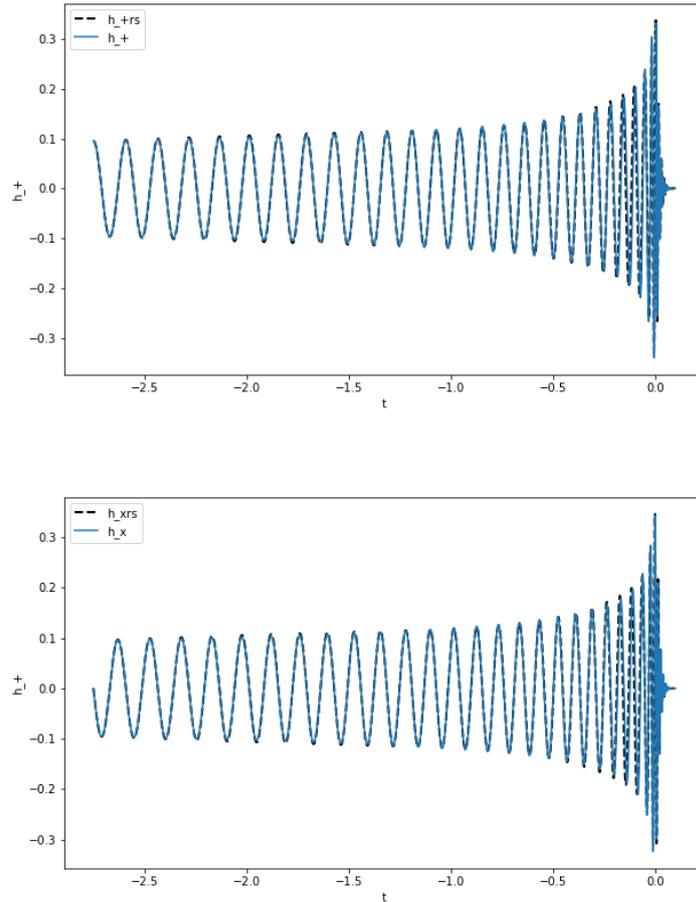


Figura 3.4: Reconstrucción de ambos perfiles de onda, $h_+(q = 2, t)$ y $h_x(q = 2, t)$, a partir de los modelos simbólicos para amplitud y fase y comparación directa con los valores de referencia. Overlap para h_+ : 0.97973554. Overlap para h_x : 0.97973539.

Resumen

En esta sección hemos mostrado que el enfoque propuesto en este trabajo produce resultados simbólicos compactos para la amplitud y fase (ecs 3.4 y 3.5, respectivamente) que resultan en un overlap de 0.979735 para las ondas simbólicas cuando son comparadas con aquellas de referencia. La descomposición de las ondas en amplitud y fase ayudó en esta tarea, aunque el ingrediente crucial usado en esta sección fue el uso de los nodos EIM.

3.3. Regresión Simbólica para todas las ondas

Pasemos a un caso más ambicioso: obtener expresiones cerradas para el rango $q \in [1, 10]^7$. A modo de ilustración, la figura 3.5 muestra las superficies en tiempo y parámetro (q) para los valores de referencia correspondientes a amplitud y fase, en

⁷Aunque este rango basta para aplicaciones astrofísicas, el procedimiento es el mismo para otros rangos.

el rango $q \in [1, 10]$ y $t \in [-2.75, 0.1]^8$.

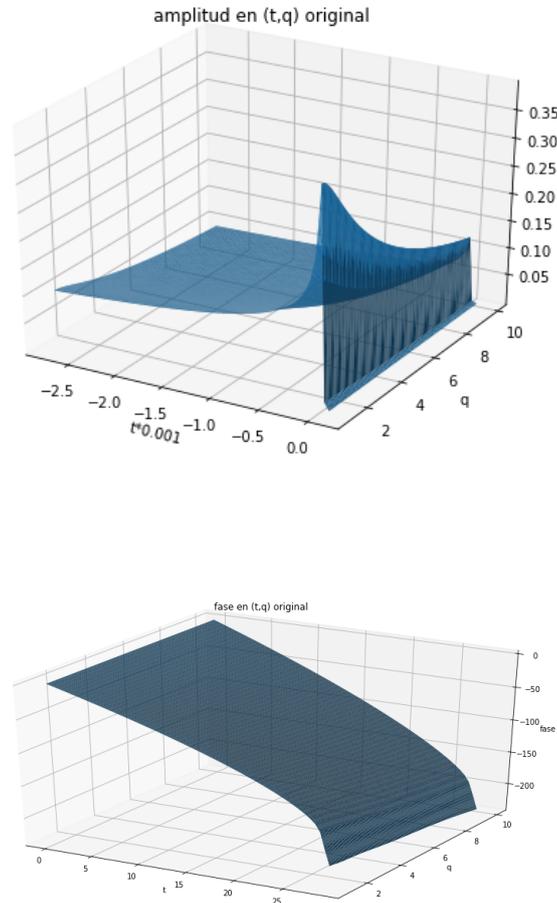


Figura 3.5: Superficies de referencia para amplitud y fase, $A(t, q)$ y $\phi(t, q)$, para $q \in [1, 10]$.

Ahora apliquemos RS para hallar expresiones cerradas para cada superficie de amplitud y fase.

Mostramos cómo un grillado adaptativo para el sampleo del dominio temporal simplifica nuevamente el problema de RS en el caso de las superficies enteras $A(t, q)$ y $\phi(t, q)$, en el plano (t, q) . La grilla se construye utilizando los nodos EIM (véase sección 5) para el tiempo en el rango $[-2.75, 0.1]$. Se utilizará indistintamente $[-2750, 100]M$ o su escala $[-2.75, 0.1]$ para referirnos al dominio temporal.

Modelos simbólicos para $q \in [1, 10]$

⁸Por razones algorítmicas de normalización que se explican más adelante, la escala temporal ha sido ajustada por un factor de 10^{-3} .

Los modelos simbólicos hallados para amplitud y fase, respectivamente, son:

$$\begin{aligned}
A(t, q) = & \{1.37502533181183 \times \exp(\operatorname{atan2}(t, 0.0409895367586908 - t))\} \\
& / \{3.40043449934568 + q - t - 1.86434379599601 \\
& \operatorname{gauss}(\operatorname{atan2}(1.1446516014466, q) - 1.49686180948812t) \\
& \operatorname{gauss}(\operatorname{atan2}(t, 0.0250835926883564 - t \\
& - 0.108134472792241tq))\} - 0.00178301085458751,
\end{aligned} \tag{3.6}$$

y

$$\begin{aligned}
\phi(t, q) = & 7.52906680521225 t^2 + 8.2690575338057 \exp(tq) - \\
& 161.936654261269 - 4.40798428725471 q - 329.774879604658 t \\
& - 1.60934544174553 tq - 0.381160938055318 \\
& \sinh^{-1}(321.674592702413 t) - 39.1334352007995 t \\
& \sinh^{-1}(321.674592702413 t)
\end{aligned} \tag{3.7}$$

donde $\operatorname{gauss}(x) := \exp(-x^2)$ y $\operatorname{atan2}(x, y)$ denota la función arcotangente de dos parámetros. Los resultados se muestran en las figuras 3.6 y 3.7.

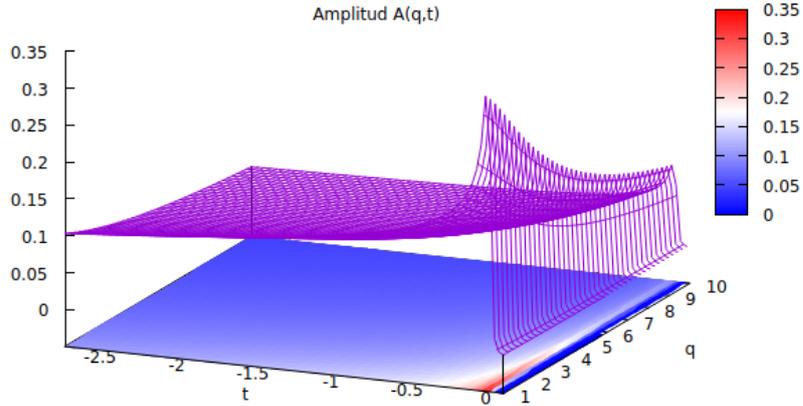


Figura 3.6: Modelo simbólico de la amplitud $A(t, q)$ para todo el dominio $[-2.75, 0.1] \times [1, 10]$ (el tiempo está escalado por 0.001), usando nodos EIM y un sampleo equiespaciado para q de 90 puntos. Coeficiente de determinación: $R^2=0.99935323$.

Hasta aquí hablamos de la misma precisión, para amplitud y fase, de la sección anterior. Pero en el momento de reconstruir las ondas h_+ y h_x para cualquier q , pequeños errores en la fase se amplifican en el tiempo, llevando a una deterioración del overlap entre modelos simbólicos *de las ondas* respecto de sus valores de referencia. Esto se detalla en el Apéndice A. En la próxima subsección presentamos una solución simple a este problema.

Resumen

Encontramos nuevamente que el uso de nodos EIM permite conseguir expresiones simbólicas de alta precisión para la amplitud y fase, no solo revelando la dependencia funcional en el tiempo sino también en el parámetro considerado (el cociente de las masas de los agujeros negros). Encontramos que al reconstruir las ondas, pequeños errores en las fases se amplifican en el tiempo, deteriorando la precisión de las ondas reconstruidas. Muestras de este deterioramiento se dan en el Apéndice A.

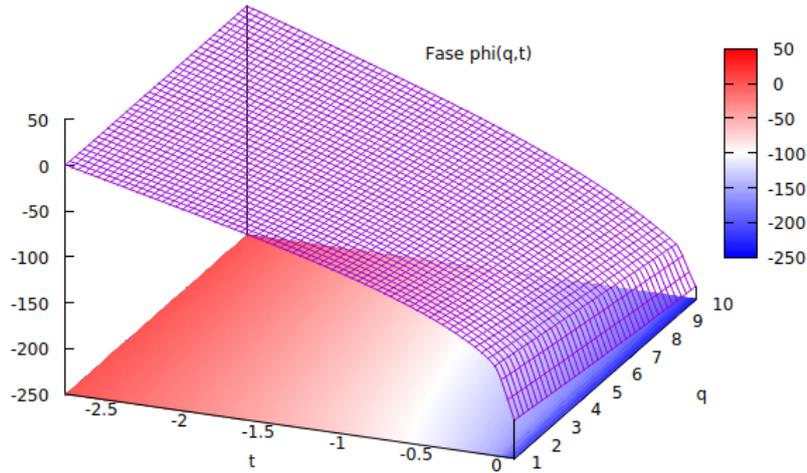


Figura 3.7: Modelo simbólico de la fase $\phi(t, q)$ para todo el dominio $[-2.75, 0.1] \times [1, 10]$, usando nodos EIM y un sampleo equiespaciado para q de resolución $\Delta q=0.5$. Coeficiente de determinación: $R^2=0.99980652$.

3.3.1. Descomposición de dominio

Para hacer frente al problema de la sensibilidad de la fase en la reconstrucción de las ondas, implementamos la siguiente solución: descomposición de dominio. Se realizó una simple descomposición uniforme en q del tipo

$$q \in [1, 2], [2, 3], [3, 4], [4, 5], [5, 6], [6, 7], [7, 8], [8, 9], [9, 10].$$

Es muy probable que descomposiciones más eficientes puedan obtenerse (por ejemplo, a través de hp-greedy [16–19]). Pero a los fines de esta tesis, este enfoque fue suficiente.

Notas:

- Esta descomposición quizás afecte la elegancia del formalismo presentado, pero descomposiciones de dominio son comunes, por ejemplo, en la resolución numérica de ecuaciones en derivadas parciales.
- Quizás haya enfoques aquí no explorados que permitan una expresión global en el espacio de parámetros.
- Desde el punto de vista de performance, esta descomposición no afecta para nada cualquier uso práctico, dado que simplemente se arma una ‘lookup table’ a la que se consulta (se hace un ‘query’) rápidamente para ver qué modelo aplicar (esto es un procedimiento standard en muchas disciplinas).
- La misma expresión global para la amplitud, ec. 3.6, es usada para todo el rango $q \in [1, 10]$ de aquí en más. Esto es quizás paradójico dado que la amplitud tiene una estructura más compleja que la fase, pero entendible teniendo en cuenta que la fase tiene una escala dos órdenes de magnitud más grande que la amplitud. Esto implica que los errores en la fase, por pequeños que sean, se propagan en la onda reconstruida de forma sensible, ya que la misma está modulada por la amplitud.

A continuación mostramos resultados para un subdominio en particular; nuevamente, para el caso general, referirse al capítulo 7.

Subdominio $q \in [1, 2]$

Mostramos la fase simbólica para el primer intervalo $[1, 2]$ de q :

$$\begin{aligned}
 \phi_{[1,2]}(t, q) = & -155.028253720751 - 273.761744720467t - 1.30615396256548q^2 \\
 & - 4.52408694055768\sinh^{-1}(21.4405231057664t) \\
 & - 0.468432096859285tq^2 - 50.7479103073014t \\
 & \operatorname{atan2}(t, -198.448702159464) \\
 & - 13.9438128055345t\sinh^{-1}(21.4405231057664t)
 \end{aligned} \tag{3.8}$$

Reconstrucción

La reconstrucción de los modos h_+ y h_x para $q = 2$ se muestran en la fig. 3.8. La diferencia con los resultados mostrados en la sección 3.2 reside en que ahora usamos la expresión para la fase (3.8), que es válida en el intervalo $q \in [1, 2]$, mientras que en la sección 3.2 la fase es válida sólo para $q = 2$. Vemos cómo las ondas son bien reconstruidas, siendo indistinguibles, por simple inspección, de las ondas de referencia. En la sección 7 cuantificamos cuán bien el modelo simbólico (no sólo para este subdominio, sino para todos) replica las ondas de referencia.

Resumen

Hemos resuelto el problema de la amplificación de errores en las fases simbólicas al reconstruir las ondas, usando una simple decomposición de dominio, mostrando un ejemplo como caso de éxito. Los resultados completos están presentados en el capítulo 7.

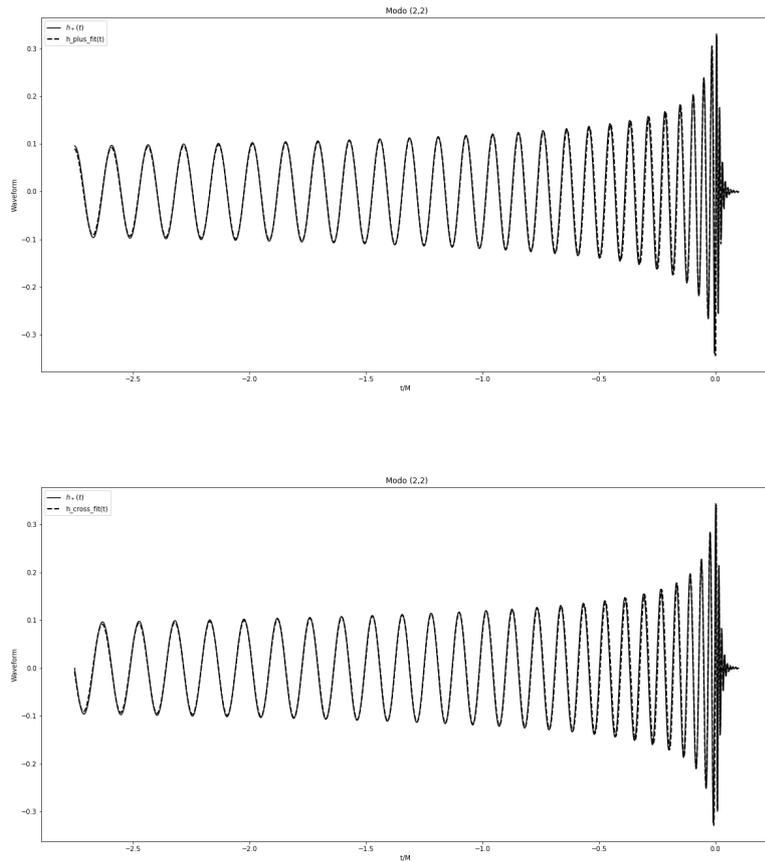


Figura 3.8: Panel superior: $h_+(q = 2, t)$. Panel inferior: $h_x(q = 2, t)$. Línea sólida: onda de referencia. Línea discontinua: onda vía modelos simbólicos. Como se puede observar a simple vista, son indistinguibles. Medidas de errores cuantitativos para todos los valores de q se presentan en sección 7.

Capítulo 4

Teoría de Aproximaciones: Aprendizaje y Bases Reducidas

La aproximación de Bases Reducidas (BR) es un marco para resolver eficientemente problemas parametrizados, representar soluciones en forma compacta y predecir otras basándonos en una descomposición *offline-online* del problema. Esto último quiere significar dividir la tarea en una parte de entrenamiento, quizá computacionalmente costosa (*offline*), y otra de evaluación, predicción y análisis, que es directa y rápida de ejecutar (*online*). La primera se implementa una sola vez para el problema de interés; la segunda, cada vez que sea necesario. Esto se hará más concreto cuando describamos el formalismo y un enfoque semi-óptimo.

4.1. Aproximación por proyección

Supongamos que nuestro espacio de interés \mathcal{F} es no lineal (es decir, la suma de dos elementos de \mathcal{F} no pertenecen necesariamente a \mathcal{F}) pero es posible embeberlo en un espacio de Hilbert arbitrario (en muchas aplicaciones de interés esto es posible). Un ejemplo concreto de no linealidad es el espacio de ondas gravitacionales, es decir, un conjunto de funciones dependientes del tiempo e indexadas por un parámetro continuo que satisfacen las ecuaciones de Einstein. Dicho espacio no es vectorial ya que, en general, la suma de dos ondas gravitacionales no es otra onda gravitacional. Motivamos este capítulo con los siguientes interrogantes: ¿es posible aproximar dicho espacio por uno vectorial, de dimensionalidad finita y alta precisión (digamos al nivel de redondeo en una computadora)? En caso afirmativo, ¿cuál es la mejor manera de hallar dicho espacio, de la forma más compacta y rápida de evaluar posible?

En esta sección introduciremos aproximaciones basadas en proyección, pero antes de entrar en detalle, establezcamos un resultado básico.

Proyectores

Consideremos un subespacio vectorial n -dimensional W_n perteneciente a un espacio de Hilbert \mathcal{H} (con producto interno $\langle \cdot | \cdot \rangle$ asociado) real o complejo. Un criterio común en teoría de aproximaciones es aquel de cuadrados mínimos. Esto es, se busca aproximar un elemento $\mathbf{v} \in \mathcal{H}$ por un $\mathbf{w} \in W_n$ solución de

$$\mathbf{w} = \operatorname{argmin}_{W_n} \|\mathbf{v} - \mathbf{w}_n\|^2, \quad (4.1)$$

donde $\mathbf{w}_n \in W_n$ y $\|\cdot\|$ es la norma L_2 inducida por el producto interno. Para resolver este problema, introducimos una base ortonormal $B_n = \{\mathbf{e}_i\}_{i=1}^n$ arbitraria de W_n .

Es fácil de demostrar que la solución óptima \mathbf{w} puede expresarse como

$$\mathbf{w} = \sum_{i=1}^n \langle i | \mathbf{v} \rangle \mathbf{e}_i. \quad (4.2)$$

También puede demostrarse que esta es una solución geométrica, i.e., que la aproximación no depende de la base. A raíz de esta observación, podemos definir un operador de proyección

$$\hat{P}_n := \sum_{i=1}^n \mathbf{e}_i \otimes \mathbf{e}_i,$$

donde \otimes denota el producto tensorial entre ambos vectores (más estrictamente, es el producto entre el vector y el covector asociado). De este modo, la solución al problema de minimización pueda expresarse como $\mathbf{w} = \hat{P}_n \mathbf{v}$.

Cuando hablamos de la mejor representación de un elemento dado de cierto espacio, lo hacemos respecto de cierto criterio de minimización. En este caso, el criterio es de cuadrados mínimos.

4.1.1. Errores colectivos

Para comenzar a responder los interrogantes del inicio del capítulo, partamos de un caso concreto. Supongamos que se quiere aproximar todo un conjunto de vectores

$$\mathcal{F} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N | \mathbf{v}_i \in \mathcal{H}\},$$

por la combinación lineal de los elementos de un cierto subespacio $W_n \in \text{Span } \mathcal{F}$. De la discusión anterior, sabemos que $\hat{P}_n \mathbf{v}_i$ es la mejor representación de un \mathbf{v}_i dado, bajo la norma inducida $\|\cdot\|$, en W_n . ¿Cómo cuantificamos el error global de dicha aproximación? Hay dos criterios usuales:

$$(1) \quad \frac{1}{N} \sum_{i=1}^N \|\mathbf{v}_i - \hat{P}_n \mathbf{v}_i\|, \quad (2) \quad \max_i \|\mathbf{v}_i - \hat{P}_n \mathbf{v}_i\|, \quad (4.3)$$

donde el primero corresponde al *error medio* y el segundo, al *error máximo*. Observemos que un criterio de aproximación (u optimización) basado en (2) es más estricto que otro basado en (1). El método de Descomposición en Valores Singulares minimiza (1) (el cual no será discutido aquí). En la siguiente sección veremos que (2) conduce a otro enfoque de Reducción de Dimensional, superior al primero: el de Bases Reducidas (BR).

Algo que surge naturalmente cuando se plantea (4.1) aplicado sobre el conjunto \mathcal{F} , es preguntarnos cuál es el mejor subespacio $W_n \subset \text{Span}(\mathcal{F})$ para llevar a cabo la aproximación. Esto conduce a un nuevo problema de optimización, discutido a continuación.

4.1.2. Medida n-Kolmogorov

En la sección anterior nos preguntamos por el mejor subespacio W_n del $\text{Span}(\mathcal{F}) \subset \mathcal{H}$, con $\mathcal{F} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N | \mathbf{v}_i \in \mathcal{H}\}$, para optimizar la aproximación (4.2). Esto nos

conduce a definir un nuevo problema, a saber, cuál es la mejor base para representar un conjunto de elementos \mathcal{F} de forma lineal. Veamos cómo formalizar esto. Dado un espacio compacto $\Omega \subset \mathcal{H}$ (no necesariamente lineal) parametrizado por una tupla $\boldsymbol{\lambda} \in \mathcal{T} \subset \mathbb{R}^m$, definimos la medida *n-Kolmogorov* (*Kolmogorov n-width*) como

$$d_n = d_n(\Omega, \mathcal{H}) := \min_{W_n \in \mathcal{H}} \max_{\boldsymbol{\lambda}} \min_{c_i \in \mathbb{C}} \left\| \mathbf{v}(\boldsymbol{\lambda}) - \sum_{i=1}^n c_i(\boldsymbol{\lambda}) \mathbf{e}_i \right\|^2, \quad (4.4)$$

donde $\{\mathbf{e}_i\}_{i=1}^n$ es una base cualquiera que expande $W_n \subset \mathcal{H}$. Sabemos que, dada una base, la mejor aproximación en el subespacio generado por ella de un vector arbitrario de \mathcal{H} viene dado por su proyección ortogonal en dicho subespacio. De este hecho se sigue que podemos reexpresar d_n de forma más geométrica como

$$d_n = d_n(\Omega, \mathcal{H}) := \min_{W_n \in \mathcal{H}} \max_{\boldsymbol{\lambda}} \left\| \mathbf{v}(\boldsymbol{\lambda}) - \hat{P}_n \mathbf{v}(\boldsymbol{\lambda}) \right\|^2. \quad (4.5)$$

Es necesario establecer aquí una observación: en la segunda instancia de la medida *n-Kolmogorov*, i.e. en el cómputo del máximo, estamos invocando el segundo criterio de aproximación definido en la sección 4.1.1, que nos llevará a formular el marco de BR más adelante.

Obsérvese que \hat{P}_n depende de W_n , por lo que puede reemplazarse el requerimiento del mejor subespacio en el problema de minimización por el del mejor operador proyector que lo lleve a cabo. Esta aclaración parece trivial, pero a nivel práctico, uno descompone los elementos en determinada base y esto, a su vez, impone cierta complejidad en el momento de pensar algoritmos que efectúen la búsqueda. En palabras, lo que esta medida dice es: agárrese el peor error de proyección en Ω , i.e. la máxima de las normas cuadradas de $\delta \mathbf{v}(\boldsymbol{\lambda}) = \mathbf{v}(\boldsymbol{\lambda}) - \hat{P}_n \mathbf{v}(\boldsymbol{\lambda})$ y búsqese el subespacio W_n que la minimice¹.

Ahora bien, este marco conceptual nos sirve para definir precisamente el problema de aproximar un cjto. de vectores por el mejor subespacio *n*-dimensional de \mathcal{H} . Para ello, necesitamos la versión discreta de la medida *n-Kolmogorov*. Discreticemos el espacio Ω y hagamos $\Omega = \mathcal{F}$. Entonces los elementos de \mathcal{F} estarán parametrizados por un índice multidimensional discreto $\boldsymbol{\lambda}_i$ perteneciente a un espacio compacto \mathcal{T} de cardinalidad finita. Ahora, para acabar de dar forma a nuestro problema, restringimos la busca del mejor subespacio que aproxime \mathcal{F} no al espacio universal \mathcal{H} , sino al *Span* \mathcal{F} . La justificación de esto se aclarará más adelante.

Comentario 1. Resaltemos algunos aspectos importantes.

- La compacidad del espacio Ω es crucial para garantizar la existencia del mínimo de los errores en la busca del mejor subespacio W_n .
- En la práctica, este problema de minimización tiene complejidad combinatorial, i.e., no computable en la práctica.
- La medida *n-Kolmogorov* representa una cota superior teórica a la precisión de cualquier procedimiento práctico que intente encontrar una solución aproximada (de ser posible casi-óptima y de una forma cuantificable) a este problema.

¹Un libro clásico sobre medidas de aproximación es [30] y una referencia más reciente, [29].

4.2. Bases Reducidas

Hasta aquí hemos discutido aproximaciones lineales por proyección ortogonal de vectores sobre el *Span* de una base. Todo esto en un ambiente general, a saber, un espacio de Hilbert arbitrario. Este es un acercamiento genérico, en el sentido de que encontramos expresión o concreción del mismo en ejemplos estándar tales como descomposición de funciones en modos de Fourier o bases polinómicas como funciones de Legendre, Hermite, Bessel, etc. Una de las diferencias más importantes respecto a aproximación ‘estándar’ es que **en este enfoque se utilizan elementos del espacio de interés como elementos de la base aproximante**. Es decir, si se quieren representar ondas gravitacionales, se usan algunas de ellas como base; si se quieren representar vacas, vacas son entonces los elementos de la base.

A fin de hacer más tangibles las cosas, y con miras a la aplicación de ondas gravitacionales, definamos nuestro espacio de Hilbert como el espacio vectorial de funciones reales h a valores complejos, esto es,

$$h : \mathbb{R}^m \rightarrow \mathbb{C}.$$

Ahora describamos nuestro *conjunto de entrenamiento* $\mathcal{F} \subset \mathcal{H}$ como el subconjunto discreto de funciones $h(\boldsymbol{\lambda}_i)$ parametrizadas con un índice real $\boldsymbol{\lambda}_i \in \mathcal{T}$, donde \mathcal{T} es el *espacio de entrenamiento*. Estas definiciones son convenientes para trasladarlas más adelante, de forma natural, al estudio de ondas gravitacionales, en el que el dominio de las h será el tiempo o la frecuencia y su valor complejo codificará los dos modos de polarización de la teoría de ondas gravitacionales. El programa de BR consiste en la búsqueda de una base de funciones cuyo *Span* aproxime un conjunto \mathcal{F} bajo cierta tolerancia previamente especificada por el usuario. Este formalismo se ajusta especialmente a problemas numéricos -computacionalmente costosos- que deben ser resueltos para construir cada función (tales como problemas de valores iniciales o de valuación en la frontera). BR es un enfoque invaluable en *Big Data*, cuando lo que se necesita es una representación poco densa (*sparse*) de la información y/u operaciones sobre ella. En general, la información de la que se dispone, los datos tangibles para manipular, son un subconjunto de un ambiente abstracto sobre el cual teorizamos y armamos modelos, pero al que solo tenemos acceso parcial únicamente a través de dicho subconjunto. Si este último es costoso de obtener, uno quisiera restringirse al mismo en el momento de efectuar algún enfoque de reducción dimensional. Esto es, quisiéramos usar solo estos datos para armar nuestra base reducida. Así, justificamos la restricción que hicimos en la sección 4.1.2 de la busca del mejor subespacio que aproxime \mathcal{F} , al $Span(\mathcal{F})$: queremos aproximar \mathcal{F} por elementos que pertenezcan al mismo conjunto.

Entonces, lo que se busca en el enfoque de BR es un conjunto minimal, casi-óptimo (en un sentido preciso),

$$B_r = \{h(\boldsymbol{\Lambda}_i)\}_{i=1}^n,$$

de funciones que pertenezcan al conjunto de interés \mathcal{F} y cuyo *Span* aproxime \mathcal{F} . Los parámetros $\boldsymbol{\Lambda}_i$ son elegidos del conjunto $\{\boldsymbol{\lambda}_i\}$ implementando algún criterio de optimalidad -que se especificará en el algoritmo a implementar-. El *Span* de la base reducida B_r , que se denotará de ahora en más por \mathbf{rb} (por *reduced basis*), corresponde al aproximante lineal de \mathcal{F} bajo cierta tolerancia, esto es

$$\mathbf{rb} := Span(B_r).$$

El producto interno se definirá a valores fijos de $\boldsymbol{\lambda}$, esto es,

$$\langle h_i, h_j \rangle := \langle h(\boldsymbol{\lambda}_i), h(\boldsymbol{\lambda}_j) \rangle,$$

respecto del $Dom(h(\boldsymbol{\lambda}))$ (dimensión física; para series temporales, t).

4.2.1. Algoritmo Greedy

Ahora introduzcamos un tipo de algoritmo para computar bases reducidas, un enfoque *greedy*, como se conoce en la literatura. Este algoritmo particular toma como entrada (*input*) el *conjunto de entrenamiento* \mathcal{F} y el *espacio de entrenamiento* \mathcal{T} , y arroja como salida (*output*) la base reducida B_r y los puntos *greedy* $\{\boldsymbol{\Lambda}_i\}$. El algoritmo, para inicializar, necesita una semilla arbitraria $\boldsymbol{\Lambda}_1$, el primer punto *greedy*, y un error de tolerancia ϵ , que funcionará como parámetro de detención del mismo. A continuación se muestra un pseudocódigo del algoritmo.

Algoritmo 1 Algoritmo greedy para bases reducidas

- 1: **Input:** $\{\boldsymbol{\lambda}_i, h(\boldsymbol{\lambda}_i)\}_{i=1}^N, \epsilon$
 - 2: Iniciar $i = 1$ y definir $\sigma_1 = 1$
 - 3: **elección de semilla** (arbitraria): $\boldsymbol{\Lambda}_1 \in \mathcal{T}$
 - 4: $\mathbf{e}_1 = h(\boldsymbol{\lambda}_1) / \|h(\boldsymbol{\lambda}_1)\|$
 - 5: $B_r = \{\mathbf{e}_1\}$
 - 6: **while** $\sigma_i \geq \epsilon$ **do**
 - 7: $i = i + 1$
 - 8: $\sigma_i(\boldsymbol{\lambda}) = \|h(\boldsymbol{\lambda}) - \hat{P}_i h(\boldsymbol{\lambda})\|^2$ con $\boldsymbol{\lambda} \in \mathcal{T}$
 - 9: $\sigma_i = \text{máx}_{\boldsymbol{\lambda} \in \mathcal{T}} \sigma_i(\boldsymbol{\lambda})$
 - 10: $\boldsymbol{\Lambda}_{i+1} = \text{argmax}_{\boldsymbol{\lambda} \in \mathcal{T}} \sigma_i(\boldsymbol{\lambda})$
 - 11: $\mathbf{e}_{i+1} = h(t; \boldsymbol{\Lambda}_{i+1}) - \hat{P}_i h(t; \boldsymbol{\Lambda}_{i+1})$ (Gram-Schmidt)
 - 12: $\mathbf{e}_{i+1} = \mathbf{e}_{i+1} / \|\mathbf{e}_{i+1}\|$ (normalización)
 - 13: $B_r = B_r \cup \mathbf{e}_{i+1}$
 - 14: **end while**
 - 15: **Output:** $B_r = \{\mathbf{e}_i\}_{i=1}^n$ y puntos *greedy* $\{\boldsymbol{\Lambda}_i\}_{i=1}^n$
-

Veamos qué es lo que dice este algoritmo. Se comienza eligiendo el primer elemento normalizado de B_r , \mathbf{e}_1 , de forma arbitraria (paso 5). En el ciclo *while* se construye el proyector $\hat{P}_1 = \mathbf{e}_1 \otimes \mathbf{e}_1$ y se computan los errores de proyección (errores *greedy*) $\sigma_1(\boldsymbol{\lambda}) = \|\delta \mathbf{v}(\boldsymbol{\lambda})\|^2 = \|h(\boldsymbol{\lambda}) - \hat{P}_1 h(\boldsymbol{\lambda})\|^2$ para todo $\boldsymbol{\lambda} \in \mathcal{T}$. Se nombra el siguiente punto *greedy* $\boldsymbol{\Lambda}_2$ como el índice de la función $h(\boldsymbol{\Lambda}_2)$ que maximiza el error de proyección hasta $\sigma_1(\boldsymbol{\Lambda}_2)$ (pasos 8 y 9). A continuación, se lleva a cabo un proceso de ortogonalización y normalización estándar, a saber, de Gram-Schmidt, para hallar el siguiente vector de B_r , \mathbf{e}_2 (pasos 10-12)². El ciclo se repite construyendo un nuevo proyector para los cálculos internos y se detiene cuando se llega a un error de proyección máximo σ_n menor o igual que el error de tolerancia ϵ , dando como resultado la base reducida $B_r = \{\mathbf{e}_i\}_{i=1}^n$ y los puntos *greedy* $\{\boldsymbol{\Lambda}_i\}_{i=1}^n$.

Comentario 1. Observaciones al Algoritmo Greedy

²El proceso de ortonormalización sigue un criterio de condicionamiento numérico.

- Según se necesite más/menos precisión en la representación BR, sólo tiene que agregarse/quitarle puntos *greedy*. Por esto se dice que los mismos son jerárquicos (i.e. anidados). Esto es importante cuando se busca economía en el costo de aprendizaje: si uno quiere mayor precisión, solo tiene que agregar más puntos a un base ya calculada. Esto es contrario a otros enfoques -como puntos de Chebyshev, de Legendre o Descomposición de Valores Singulares- que no son jerárquicos y en los que hay que comenzar el proceso desde el principio cada vez que se requiere mayor precisión.
- Cada paso *greedy* (paso 8) es *embarazosamente paralelo*, i.e., los errores *greedy* $\sigma(\boldsymbol{\lambda})$ pueden computarse simultánea e independientemente para cada $\boldsymbol{\lambda} \in \mathcal{T}$.
- Presenta complejidad constante por elemento de la base elegida y, por consiguiente, lineal en el número total de elementos de la base; esto es, el costo de agregar un nuevo elemento a B_r no depende de i , ya que

$$\hat{P}_{i+1} = \hat{P}_i + \mathbf{e}_{i+1} \otimes \mathbf{e}_{i+1}.$$

Si los coeficientes de proyección para i bases (que son números, i.e., de bajo costo de almacenamiento) se guardan en la memoria de la computadora, sólo se necesita calcular el producto interno entre las $h(\boldsymbol{\lambda})$ del conjunto de entrenamiento y \mathbf{e}_{i+1} para obtener el nuevo coeficiente de proyección.

Habíamos dicho anteriormente que la medida *n-Kolmogorov* es una cota superior teórica a la calidad de cualquier algoritmo que modele el problema de medida *n-Kolmogorov*. El algoritmo *greedy* es uno de estos que arrojan una solución casi-óptima al problema de medida *n-Kolmogorov* [7], como se describirá a continuación.

4.2.2. Convergencia del método greedy

Analicemos qué tan bien resuelve el algoritmo *greedy* el problema de medida *n-Kolmogorov*.

Sea ϵ cierta tolerancia especificada por el usuario que garantizará que, al final del ciclo del algoritmo *greedy*, se obtenga una base reducida cuyo error *greedy* total

$$\sigma_n := \max_{\boldsymbol{\lambda} \in \mathcal{T}} \sigma_i(\boldsymbol{\lambda}), \quad (4.6)$$

sea tal que

$$\sigma_n \leq \epsilon. \quad (4.7)$$

El error *greedy* total depende, naturalmente, de la base. Geométricamente, el proyector \hat{P}_n induce un subespacio lineal o hiperplano en \mathcal{H} . En el problema de medida *n-Kolmogorov*, la última instancia de la medida busca el hiperplano que minimice el error máximo de proyección. Por otro lado, el algoritmo *greedy* comienza con una semilla arbitraria como primer elemento de la base B_r (paso 3)³. Esto implica que la base reducida depende de la semilla y, por consiguiente, también el hiperplano inducido. Este "grado de libertad" en el algoritmo supone algún costo en la representatividad del

³Siendo un algoritmo de optimización global, la elección de la semilla es irrelevante. Ver [12] para una discusión explícita de este punto.

problema original. Entonces nos preguntamos cuán sub-óptimo es el método *greedy* para solucionar el problema de hallar el mejor hiperplano que minimice el máximo error de proyección.

En [7] se demostraron varios resultados poderosos: para una escala de decaimiento polinomial de la medida *n-Kolmogorov*

$$d_n \leq Cn^{-\alpha}, \quad (4.8)$$

entonces

$$\sigma_n \leq C'_\alpha n^{-\alpha} \quad \forall n \geq 1, \quad (4.9)$$

para un $\alpha > 0$ fijo.

También fue demostrado en [7] un resultado análogo para decaimiento exponencial, i.e., si

$$d_n \leq C \exp(-an^\alpha), \quad (4.10)$$

entonces

$$\sigma_n \leq C'_\beta \exp(-b_\beta n^\beta) \quad \forall n \geq 0 \quad (4.11)$$

con

$$\beta = \frac{\alpha}{\alpha + 1}.$$

La convergencia de d_n depende de la suavidad/regularidad de las funciones en el espacio de entrenamiento respecto del parámetro λ . Entonces, para problemas con dependencia paramétrica suave, se espera rápida convergencia en términos de bases *greedy* para cierta tolerancia ϵ elegida. Puede entenderse esta condición con una analogía de teoría espectral. Cuando se efectúa una descomposición espectral de una función determinada, e.g. modos de Fourier, la representabilidad de la misma en términos de un cjto. finito de elementos de la base depende de la analiticidad de la función. Puede demostrarse que si una función h es suave ($h \in C^\infty$), se obtiene *convergencia espectral*, es decir, que el error de proyección

$$\|h - \hat{P}_n h\| \quad (4.12)$$

decae más rápidamente que cualquier ley de potencias con $n \rightarrow \infty$. Asimismo, si la dependencia en el parámetro es analítica, puede probarse decaimiento exponencial de la forma $\exp(-n^{1/d})$; si ésta dependencia es C^s , entonces la convergencia es de la forma $n^{-s/d}$. Esto implica que para problemas parametrizados con cierta regularidad respecto de variación de parámetros, la medida *n-Kolmogorov*, aún si es desconocida, converge rápidamente y, por extensión, la representación de una base reducida usando un método *greedy*.

Esta es una de las características más importantes de Bases Reducidas. En sam- pleos aleatorios -tales como Monte Carlo- en los que no se explota ninguna regularidad presente, el error de la representación decae de forma sublineal, mientras que, si la dependencia respecto de variación de parámetros es C^∞ , el decaimiento es exponencial. De modos que no resulta sorprendente que el enfoque sea tan eficiente y compacto, tanto para análisis de datos como para modelos predictivos.

Capítulo 5

Modelos Predictivos

5.1. De Proyección a Interpolación

En la sección anterior vimos que, dado un subespacio W_n del espacio de Hilbert \mathcal{H} , una representación óptima en una norma L_2 de cualquier vector \mathbf{v} de \mathcal{H} es $\hat{P}_n \mathbf{v}$. Si tenemos una base ortonormal $B_n = \{\mathbf{e}_i\}_{i=1}^n$ de W_n entonces

$$\hat{P}_n \mathbf{v} = \sum_{i=1}^n \langle \mathbf{e}_i | \mathbf{v} \rangle \mathbf{e}_i \quad (5.1)$$

Los coeficientes de proyección exigen que la función sea conocida, en principio, en todo el dominio de integración. Esto puede presentar un inconveniente en cuanto costo computacional o disponibilidad de memoria cuando la cantidad de datos crece. Entonces, dando un paso más en este camino de compresión informacional, quisiéramos hacerlo ahora **no solo** en el espacio de parámetros \mathcal{T} (dominio de aplicación del formalismo de BR) sino **también** en la dimensión *física* (e.g. espacio, tiempo) de las funciones $h(\boldsymbol{\lambda})$. En este trabajo nos interesan series temporales, por lo que la dimensión física será el tiempo t .

Ahora bien, para hacer frente a este problema, proponemos aproximar una función $h(\boldsymbol{\lambda})$ por una combinación lineal de elementos de B_r , cuyos coeficientes sean solución de un problema de interpolación, a saber:

$$\mathcal{I}_n[h](t, \boldsymbol{\lambda}) := \sum_{i=1}^n C_i(\boldsymbol{\lambda}) \mathbf{e}_i(t), \quad (5.2)$$

tal que

$$\mathcal{I}_n[h](T_i, \boldsymbol{\lambda}) = h(T_i, \boldsymbol{\lambda}), \quad \forall i = 1, \dots, n \quad (5.3)$$

para cierto conjunto $T := \{T_i\}_{i=1}^n$ de nodos temporales estratégicamente elegidos -en breve explicaremos un método casi-óptimo de elección de estos nodos. La razón de elegir n nodos (tantos como elementos de la base reducida) se hará evidente cuando describamos el método.

La ecuación (5.3) es equivalente al sistema de ecuaciones lineales $n \times n$

$$\sum_{i=1}^m V_{ji} C_i(\boldsymbol{\lambda}) = h(T_j, \boldsymbol{\lambda}), \quad j = 1, \dots, n \quad (5.4)$$

para los coeficientes $C_i(\boldsymbol{\lambda})$, donde definimos la matriz de interpolación V como

$$V := \begin{pmatrix} \mathbf{e}_1(T_1) & \mathbf{e}_2(T_1) & \cdots & \mathbf{e}_n(T_1) \\ \mathbf{e}_1(T_2) & \mathbf{e}_2(T_2) & \cdots & \mathbf{e}_n(T_2) \\ \mathbf{e}_1(T_3) & \mathbf{e}_2(T_3) & \cdots & \mathbf{e}_n(T_3) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{e}_1(T_n) & \mathbf{e}_2(T_n) & \cdots & \mathbf{e}_n(T_n) \end{pmatrix}, \quad (5.5)$$

una generalización de la matriz de Vandermonde en interpolación polinomial.

La solución puede expresarse formalmente como

$$C_i = \sum_{j=1}^n (V^{-1})_{ij} h(T_j, \boldsymbol{\lambda}) \quad (5.6)$$

donde se supone la invertibilidad de la matriz V está relacionada con la independencia lineal de las columnas que forman la matriz. Se sigue de esto que el interpolante puede escribirse como

$$\mathcal{I}_n[h](t, \boldsymbol{\lambda}) = \sum_{j=1}^n \mathbf{B}_j(t) h(T_j, \boldsymbol{\lambda}) \quad (5.7)$$

donde

$$\mathbf{B}_j(t) := \sum_{i=1}^n \mathbf{e}_i(t) (V^{-1})_{ij} \quad (5.8)$$

es independiente de $\boldsymbol{\lambda}$.

Comentario 1. Observaciones.

- Este proceso conduce a una descomposición/aproximación afín de problemas que en principio no lo son, separando dependencia de parámetros del tiempo, en este caso, ver (5.7): los coeficientes \mathbf{B}_j dependen solo del tiempo, mientras que los $h(T_j, \boldsymbol{\lambda})$ lo hacen solo de parámetros (los nodos T_i son precomputados offline por única vez). Esto tiene múltiples ventajas, por ejemplo, permite el diseño de *Reduced Order Quadratures*[39].
- Dicho de otro modo, (5.7) es una combinación lineal de la función $h(\boldsymbol{\lambda})$ valuada en los nodos interpolantes T_i . Con esto logramos reducir la información necesaria (en el dominio del tiempo) para construir un representante de la función $h(\boldsymbol{\lambda})$.
- Los coeficientes \mathbf{B}_i son computados directamente de la base reducida. Esto provee una separación *offline/online* limpia: los B_i pueden ser calculados *offline* una vez que se tiene la base reducida, mientras que el interpolante $\mathcal{I}_n[h](t, \boldsymbol{\lambda})$ se evalúa de forma rápida en la fase *online* vía ecuación (5.7), cuando el parámetro $\boldsymbol{\lambda}$ es especificado por el usuario.
- Todavía se necesita conocer $h(t, \boldsymbol{\lambda})$ en los nodos interpolantes $t = T_i$ para cada $\boldsymbol{\lambda}$ y así poder construir el interpolante en (5.7). Recordemos que al principio, para construir la base reducida B_r solo poseíamos un espacio de funciones $h(t, \boldsymbol{\lambda})$ discreto en el parámetro $\boldsymbol{\lambda}$ para entrenamiento. Ahora llegamos a un método que nos dice cómo construir un modelo que aproxime dicho espacio con una precisión arbitraria; pero que **no nos dice nada de las funciones que están por fuera**

del conjunto de entrenamiento. Teniendo en mente la última observación, en la siguiente sección, luego de haber explicado cómo elegir un buen conjunto de nodos T , llevaremos este método al siguiente nivel: el predictivo.

5.2. Método de Interpolación Empírica (EIM)

Queda por discutir la elección de los nodos T_i . Este problema no es trivial, ya que, en el caso particular de interpolación polinómica, cuando T es elegido de forma equiespaciada, puede aparecer el clásico fenómeno de Runge, en el que la densificación de puntos en el dominio de interpolación, sorprendentemente, lleva a divergencias en el polinomio interpolante [34]. Aquí es donde introducimos el Método de Interpolación Empírica [26](EIM, por sus siglas en inglés) como un procedimiento para identificar un buen conjunto de nodos interpolantes T . Antes de entrar en detalles técnicos, remarquemos algunas propiedades del EIM:

- Es jerárquico (anidado). Los ‘puntos más relevantes’ en la dimensión física y los interpolantes asociados, son elegidos y construidos, respectivamente. Este proceso es dual a la construcción de la base reducida usando el algoritmo *greedy*. Es decir, por cada elemento de la base reducida, hay un nodo EIM asociado.
- El costo de calcular los nodos y construir el interpolante empírico es muy bajo, dado que ya no se trabaja con un espacio de entrenamiento, sino con la base reducida.
- Es altamente preciso. Por diseño, el algoritmo minimiza la constante de Lebesgue en cada paso del ciclo.

5.2.1. Algoritmo EIM

A continuación describimos el algoritmo para la obtención de los nodos empíricos.

Algoritmo 2 EIM

- 1: **Input:** $B_r = \{\mathbf{e}_i\}_{i=1}^n, \{t_i\}_{i=1}^L$
 - 2: $i = \operatorname{argmax}|\mathbf{e}_1|$ (argmax retorna el índice temporal de su argumento más grande).
 - 3: Hacer $T_1 = t_i$
 - 4: **for** $j = 2 \rightarrow n$ **do**
 - 5: Construir $\mathcal{I}_{j-1}[\mathbf{e}_j](t)$
 - 6: $\mathbf{r}_j = \mathcal{I}_{j-1}[\mathbf{e}_j](t) - \mathbf{e}_j$ (aquí \mathbf{r} denota el residuo).
 - 7: $i = \operatorname{argmax}|\mathbf{r}_j|$
 - 8: $T_j = t_i$
 - 9: **end for**
 - 10: **Output:** nodos EIM $\{T_i\}_{i=1}^n$
-

En palabras, el algoritmo tiene como entrada la base reducida B_r y un grillado temporal $\{t_i\}_{i=1}^L$ que genera el espacio de entrenamiento sobre el cual desarrollar el algoritmo. El primer nodo interpolante (pasos 2 y 3) se elige como el tiempo en que

el módulo del vector e_1 es máximo (los vectores son funciones). Se accede a un ciclo donde se construye primero el interpolante del siguiente elemento de la base B_r con los nodos empíricos previos, se calcula el residuo (paso 6) correspondiente, i.e., el vector error de interpolación $r_j(t)$ y se elige como siguiente nodo empírico a aquel tiempo que maximiza el módulo del residuo.

El algoritmo está diseñado para que los nodos EIM sean distintos dos a dos, como criterio elemental para evitar que la matriz interpolante sea singular: los ceros de la función residuo $r_j(t) = \mathcal{I}_{j-1}[e_j](t) - e_j$ son, por definición, los T_i y el máximo valor de $|r_j(t)|$ se da en los puntos extremos del residuo. Así que, a menos que algún interpolante sea igual al elemento de la base al que interpola (lo cual no puede suceder debido a la independencia lineal de la base), todos los nodos EIM son distintos.

5.2.2. Condicionamiento y Convergencia del EIM

Ahora estudiemos el condicionamiento numérico del EIM. Vimos que la solución formal al problema de interpolación (ec. (5.3)) implica invertir la matriz interpolante V . En análisis numérico, donde se trabaja con errores de punto flotante, no sólo es importante que la matriz sea no singular, sino que esté bien condicionada, esto es, que la amplificación de errores en un problema determinado sea lo menor posible. Para estudiar el condicionamiento de una matriz A no singular, se define un número de condicionamiento, $k(A)$, que cuantifica cuán bien o mal condicionada es una matriz. Para ser más concretos, supongamos que se quiere resolver un sistema lineal $Ax = y$ para A , y conocidos. Queremos saber cómo se propaga un error determinado δy en la solución $x = A^{-1}y$. Claramente, el error en la solución es $\delta x = A^{-1}\delta y$. Ahora definamos el número de condicionamiento $k(A)$ de la matriz A como la razón máxima entre la norma del error relativo en la solución, δx , y la norma del error relativo en y , δy . Más precisamente,

$$k(A) := \max_{y, \delta y} \frac{\|\delta x\|/\|x\|}{\|\delta y\|/\|y\|} = \max_{\delta y} \frac{\|A^{-1}\delta y\|}{\|\delta y\|} \max_x \frac{\|Ax\|}{\|x\|} \quad (5.9)$$

donde $y, \delta y \neq 0$. En la última igualdad de la ec. (5.9), encontramos las definiciones de norma matricial inducida, i.e.,

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}, \quad x \in \mathbb{R}^m$$

Con esa definición, el número de condicionamiento queda definido como $k(A) = \|A^{-1}\| \|A\|$. Obsérvese que este número depende únicamente de la estructura de la matriz, no del vector y ni su error asociado δy . Claramente representa una medida de cuán bien comportada es dicha matriz para propagar los errores en un problema tipo $Ax = y$. Un $k(A)$ muy grande amplifica el error relativo del input y .

Ahora bien, retornando a la matriz interpolante V y el problema que plantea la ec. (5.3), vemos que su condicionamiento depende de la elección de los nodos T_i . Por consiguiente, se plantea el problema de minimizar $k(V)$ para cierto cjto. de nodos temporales T . Este problema se ha estudiado en la literatura para matrices tipo Vandermonde, usando polinomios ortonormales [23]. El EIM conduce a una elección de nodos que hace que la matriz interpolante V esté bien condicionada [26].

Veamos cómo destilar del algoritmo EIM el hecho de que el mismo evita que los nodos empíricos hagan de V singular. Debido a la precisión de máquina, matrices casi-singulares están mal condicionadas numéricamente. Se busca, por tanto, que las filas y columnas de V sean lo más distintas posible en cada paso de la construcción del cjo. T . En el caso de las filas, de (5.5) vemos que los nodos T_i deben ser lo más distintos posible; en el de las columnas, que dos funciones de la base reducida no coincidan en los nodos empíricos, i.e., que $\mathbf{e}_i(T_k) \neq \mathbf{e}_j(T_k) \forall i, j, k$. Veamos cómo el algoritmo EIM asegura estas dos condiciones simultáneamente. Sigamos los primeros pasos del algoritmo e intentemos ver que dos filas o columnas no pueden coincidir nunca. Partiendo de T_1 , el primer interpolante del ciclo es

$$\mathcal{I}_1[\mathbf{e}_1](t) = \frac{\mathbf{e}_2(T_1)}{\mathbf{e}_1(T_1)} \mathbf{e}_1(t) \quad (5.10)$$

y el residuo asociado,

$$\mathbf{r}_2(t) = \frac{\mathbf{e}_2(T_1)}{\mathbf{e}_1(T_1)} \mathbf{e}_1(t) - \mathbf{e}_2(t). \quad (5.11)$$

Ahora, para construir el siguiente interpolante, debe invertirse la matriz

$$V := \begin{pmatrix} \mathbf{e}_1(T_1) & \mathbf{e}_2(T_1) \\ \mathbf{e}_1(T_2) & \mathbf{e}_2(T_2) \end{pmatrix}. \quad (5.12)$$

Veamos que la elección de T_2 en el algoritmo EIM garantiza su buen condicionamiento: si resultase $T_2 = T_1$ (caso de filas iguales en la matriz V) en la busca del $\mathit{argmax}|\mathbf{r}_2(t)|$, tendríamos $\mathbf{r}_2(T_2) = 0$; como $|\mathbf{r}_2(t)| \geq 0$, este resultado implica que $\mathbf{r}_2(t) = 0 \forall t$ y, por consiguiente, $\mathcal{I}_1[\mathbf{e}_1](t) = \mathbf{e}_2(t) \forall t$, en contradicción con el supuesto de independencia lineal de la base reducida. Ergo, $T_2 \neq T_1$. Por otro lado, supongamos que $\mathbf{e}_1(T_1) = \mathbf{e}_2(T_1)$ y $\mathbf{e}_1(T_2) = \mathbf{e}_2(T_2)$ (caso de columnas iguales), entonces

$$\mathbf{r}_2(t) = \mathbf{e}_1(t) - \mathbf{e}_2(t). \quad (5.13)$$

Pero el máximo de $|\mathbf{r}_2(t)|$ se alcanza en T_2 y, por hipótesis, $\mathbf{e}_1(T_2) = \mathbf{e}_2(T_2)$. Esto implica que $\mathit{máx}|\mathbf{r}_2(t)| = 0$ para $t = T_2$ y, otra vez, que $\mathbf{r}_2(t) = 0 \forall t$, es decir, $\mathbf{e}_1(t) = \mathbf{e}_2(t) \forall t$, en contradicción con la independencia lineal de ambos vectores.

Este argumento puede extenderse paso a paso en el algoritmo, demostrando que los nodos empíricos se construyen de manera que V sea no singular. El hecho de que, en sentido numérico, el algoritmo tiende a hacer de V lo menos singular posible se ve en que la elección de los nodos T_i se lleva a cabo de manera que estos se distribuyan lo más alejados posible unos de otros en la recta real. Veamos por qué:

Si reexpresamos la ec. (5.11) como

$$\mathbf{r}_2(t) = \frac{\mathbf{e}_2(T_1)\mathbf{e}_1(t) - \mathbf{e}_2(t)\mathbf{e}_2(T_1)}{\mathbf{e}_1(T_1)}, \quad (5.14)$$

vemos que el numerador es el determinante de la matriz (5.12) a menos de un signo, con T_2 reemplazado por t . Si denotamos este determinante como $\Delta_2(t)$, entonces el problema de maximizar $|\mathbf{r}_2(t)|$ es equivalente al de maximizar $|\Delta_2(t)|$, ya que $\mathbf{e}_2(T_1)$ es una cte. Esta observación es completamente general, i.e.,

$$|\mathbf{r}_j(t)| = \frac{|\Delta_j(t)|}{|\Delta_{j-1}(T_{j-1})|} \quad \forall j \quad (5.15)$$

Entonces es claro. Si se quiere alejar la posibilidad de singularidad, es necesario que el determinante de la matriz interpolante sea lo más distinto de cero posible (equivalente a pedir que sus filas y columnas sean lo más disímiles posible). Esto implica que los T_i deben estar alejados unos de otros para garantizar que V sea invertible.

El análisis de convergencia del interpolante se sigue, por definición, de la convergencia del algoritmo *greedy*. Sin entrar en detalles (véase [39]), decimos que

$$\|h(\boldsymbol{\lambda}) - \mathcal{I}_n[h(\boldsymbol{\lambda})]\|^2 \leq \Lambda_n \|h(\boldsymbol{\lambda}) - \hat{P}_n h(\boldsymbol{\lambda})\|^2, \quad (5.16)$$

donde

$$\Lambda_n := \|\mathcal{I}_n\|^2 = \max_{\|h(\boldsymbol{\lambda})\|=1} \|\mathcal{I}_n[h(\boldsymbol{\lambda})]\|^2 \quad (5.17)$$

se conoce como *constante de Lebesgue*. Si en la ec. (5.16) tomamos el máximo en $\boldsymbol{\lambda}$ en ambos miembros, obtenemos

$$\max_{\boldsymbol{\lambda}} \|h(\boldsymbol{\lambda}) - \mathcal{I}_n[h(\boldsymbol{\lambda})]\|^2 \leq \Lambda_n \max_{\boldsymbol{\lambda}} \|h(\boldsymbol{\lambda}) - \hat{P}_n h(\boldsymbol{\lambda})\|^2, \quad (5.18)$$

donde reconocemos, de la sección 4, el error *greedy*

$$\sigma_n := \max_{\boldsymbol{\lambda}} \|h(\boldsymbol{\lambda}) - \hat{P}_n h(\boldsymbol{\lambda})\|^2. \quad (5.19)$$

Entonces podemos escribir la ec. (5.18) como

$$\max_{\boldsymbol{\lambda}} \|h(\boldsymbol{\lambda}) - \mathcal{I}_n[h(\boldsymbol{\lambda})]\|^2 \leq \Lambda_n \sigma_n, \quad (5.20)$$

Comentario 1. Subrayemos algunos puntos importantes.

- La constante de Lebesgue Λ_n [6] tiende a crecer con n . En interpolación polinomial, su crecimiento es exponencial para nodos equiespaciados y logarítmico para nodos de Chebyshev, usando la norma infinita $\|h\|_\infty := \max_t |h|$ [34]. Los nodos de EIM emulan el comportamiento de estos últimos para problemas especializados (i.e. bases adaptadas al problema de interés).
- El carácter decreciente del error *greedy* σ_n , emulando la convergencia espectral de análisis matemático -exponencial si el campo de funciones $h(\boldsymbol{\lambda})$ subyacente al cjto. de entrenamiento \mathcal{F} presenta suavidad en la variación de parámetros $\boldsymbol{\lambda}$ - implica que, en el caso de una dimensión, los nodos de Chebyshev son una elección óptima para que el error máximo del interpolante esté acotado por una cantidad decreciente (Λ_n (crecimiento logarítmico) \times σ_n (decaimiento exponencial)). El problema que presentan estos nodos es el de no ser anidados y no estar adaptados a problemas de interés. Es decir, si se quiere alcanzar mayor precisión, los nodos de Chebyshev deben ser computados desde cero en cada caso. Esto se resuelve con el algoritmo EIM, cuyos nodos emulan, de un modo casi-óptimo, a los de Chebyshev, con la ventaja de ser anidados: mayor precisión implica agregar más nodos a los ya calculados.

5.3. Modelos sustitutos

El siguiente paso es generar un modelo predictivo en función de los datos comprimidos. Observemos que, para construir el interpolante $\mathcal{I}_n[h(\boldsymbol{\lambda})](t)$ de una función $h(t, \boldsymbol{\lambda})$, es necesario conocerla en los nodos empíricos T_i y el parámetro $\boldsymbol{\lambda}$. Recordemos que nuestro campo de conocimiento primordial se corresponde con el conjunto discreto de entrenamiento \mathcal{F} . De esto inferimos que sólo podemos aproximar funciones que estén en \mathcal{F} . En muchos casos de interés, \mathcal{F} es un representante parcial de un espacio de funciones subyacente cuya dependencia en el parámetro $\boldsymbol{\lambda}$ es suave (e.g. ondas gravitacionales). Por lo que es natural preguntarse por un modelo que no solo replique \mathcal{F} , sino que anticipe al modelo subyacente. Para ello necesitamos de un método genérico que "llene los espacios vacíos", con el objeto de construir funciones que sustituyan al modelo real o subyacente con un grado alto de fidelidad. A continuación, se describe cómo construir un modelo sustituto que involucre combinar todo lo expuesto anteriormente en un solo marco teórico.

Etapa Offline

- 1) Seleccionar, vía algoritmo *greedy*, los n puntos en el espacio de parámetros \mathcal{T} , i.e. los $\boldsymbol{\lambda}_i$ que servirán para construir la base reducida.
- 2) Identificar los n nodos empíricos T_i en la dimensión *física* (tiempo) vía algoritmo EIM.
- 3) Para cada nodo empírico, realizar un fiteo de la función $h_i(\boldsymbol{\lambda}) := h(T_i, \boldsymbol{\lambda})$ (e.g. polinomial) en la dimensión paramétrica **utilizando los puntos *greedy*** del paso 1), obteniendo así un sustituto $h_i^s(\boldsymbol{\lambda})$ de la función $h_i(\boldsymbol{\lambda})$ en cada nodo T_i .

Etapa Online

- 4) Construir la función sustituta $h^s(t, \boldsymbol{\lambda})$ vía el interpolante empírico

$$h^s(t, \boldsymbol{\lambda}) := \sum_{i=1}^n \mathbf{B}_i(t) h^s(T_i, \boldsymbol{\lambda}) \quad \forall t, \boldsymbol{\lambda}. \quad (5.21)$$

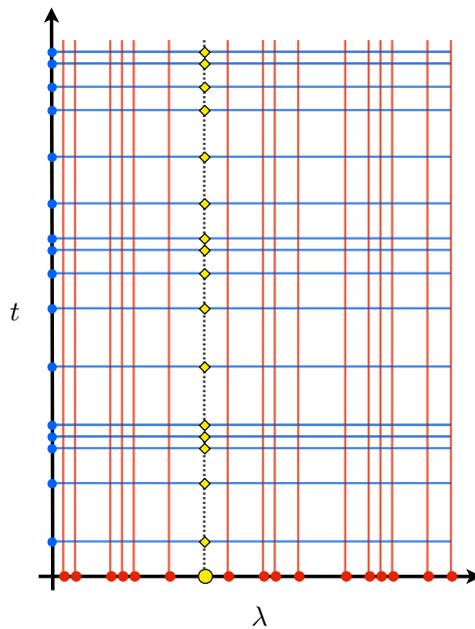


Figura 5.1: Representación esquemática de la construcción y evaluación del modelo predictivo. **Puntos rojos**: puntos greedy en el espacio de parámetros (paso 1, offline). **Puntos azules**: (paso 2, offline) nodos EIM. **Líneas azules**: (paso 3, offline) fiteo de la función $h_i(\boldsymbol{\lambda})$ en el nodo empírico T_i . El **punto amarillo** es un parámetro $\boldsymbol{\lambda}$ genérico que representa las predicciones -**diamantes amarillos**- del modelo en cada nodo EIM. **Línea negra punteada**: construcción de la función sustituta h_s para todo tiempo, vía el interpolante empírico (paso 4, online).

Capítulo 6

Programación Genética

6.1. Introducción

Programación genética (PG) es, en breve, un campo de Inteligencia Artificial cuyo principal objetivo es la evolución de programas de computadora. Las técnicas implementadas en PG emulan las formas de la Naturaleza, i.e., se modelan algoritmos que replican, o pretenden hacerlo, el proceso de evolución natural que se da en la misma. De esta manera, se migran de forma natural los conceptos usados en biología al ámbito de la computación. Las dos referencias estándar en PG son [25] (la ‘enciclopedia’ en el campo) y [32] (una visión general).

El esquema que siguen todos los algoritmos de PG puede condensarse en el siguiente pseudocódigo:

Algoritmo 3 Programación Genética

- 1: **Crear** estocásticamente una *población inicial* de programas.
 - 2: **Repetir**
 - 3: *Ejecutar* cada programa y computar su calidad o *fitness*.
 - 4: Seleccionar uno o dos programas de la población con una probabilidad basada en su/sus fitness para participar en operaciones genéticas.
 - 5: Crear nuevos programas mediante la aplicación de operaciones genéticas.
 - 6: **Hasta que** se haya alcanzado una solución aceptable o satisfecho alguna condición de detención (e.g. máximo número de generaciones alcanzado).
 - 7: **Retornar** el/los mejor/res individuo/s.
-

Comentario 1. Remarquemos algunos puntos del pseudocódigo:

- En el paso 1) se asume la existencia de un entorno de primitivas con las que formar la población inicial.
- En el paso 2), computar el fitness de cada programa refiere a tener un método cuantitativo para valuar los programas de una generación dada, i.e., asignarle determinado peso según los requerimientos del usuario (*presión evolutiva* sobre los individuos).

- En el paso 3) y 4) se seleccionan aquellos programas mejor valuados según su fitness para producir nuevos programas a partir de operaciones genéticas, e.g., *mutación (mutation)* (alterar al azar una parte de un programa para crear otro nuevo); *cruzamiento (crossover)* (combinación de partes elegidas al azar de cada programa de un par seleccionado para producir otro). Más adelante veremos un ejemplo concreto de PG donde estos conceptos se harán tangibles.
- El ciclo se repite produciendo, en cada paso, una generación (probablemente) más óptima que la anterior, hasta satisfacer cierto criterio de detención.
- Hay que remarcar la importancia de la *estocasticidad* en la producción de nuevos programas, ya que, como en selección natural, los individuos se ven sujetos a distintas presiones evolutivas (fitness) que actúan en sinergia con cambios azarosos en la constitución genética de los mismos. El azar en PG, como en la Naturaleza, hace que mejores resultados puedan no ser alcanzados en el paso de una generación a otra pero, a su vez, permite la evasión de trampas en las que métodos deterministas son susceptibles de caer. Siguiendo esta línea, un aspecto importante de PG reside en que nunca se descarta un individuo con mal fitness: esto es una diferencia grande respecto de algoritmos no evolutivos.

6.2. Regresión Simbólica (RS)

Ahora introducimos un método particular de PG llamado Regresión Simbólica (RS), para la busca de expresiones matemáticas que mejor representen determinado conjunto de datos. El objetivo es evolucionar relaciones simbólicas hasta haber satisfecho cierto criterio de optimalidad que nos permita cuantificar cuán bueno es el modelo respecto de los datos introducidos. A diferencia de otros enfoques, en los que la forma específica de la solución es determinada por el usuario -e.g. regresión lineal o polinomial-, en RS -en su forma más sofisticada- no lo es: implica una búsqueda no supervisada, libre de bias humano.

En RS se parte de un diccionario arbitrario de funciones o primitivas como arena para combinar y producir nuevas expresiones y relaciones. Debe definirse una sintaxis como plataforma para definir y producir expresiones. En general, en este campo se implementa una sintaxis de árbol para representar expresiones matemáticas. Una vez la generación de expresiones está lista, se aplica una función *fitness* que evalúa la optimalidad de las expresiones formadas (e.g. minimizar el error máximo que existe entre el modelo y los datos o maximizar la correlación, etc.; la noción de error es arbitraria) y, acorde a la jerarquía de dicha evaluación, se producen nuevas expresiones que tienden a optimizar el modelo. Esta función es el criterio evolutivo que actúa de *presión ambiental* sobre los individuos o expresiones de una generación.

Veamos un ejemplo ilustrativo del mecanismo de RS. En el diagrama 6.1¹, los nodos reciben el nombre de *función*, mientras que cada variable se denomina *terminal*. Se observa en él el funcionamiento de la operación genética *cruzamiento* para producir una nueva expresión.

Existe una dinámica delicada entre capacidad computacional y configuración de los parámetros evolutivos en función del algoritmo de PG a utilizar. La complejidad en

¹Imagen extraída de [32]

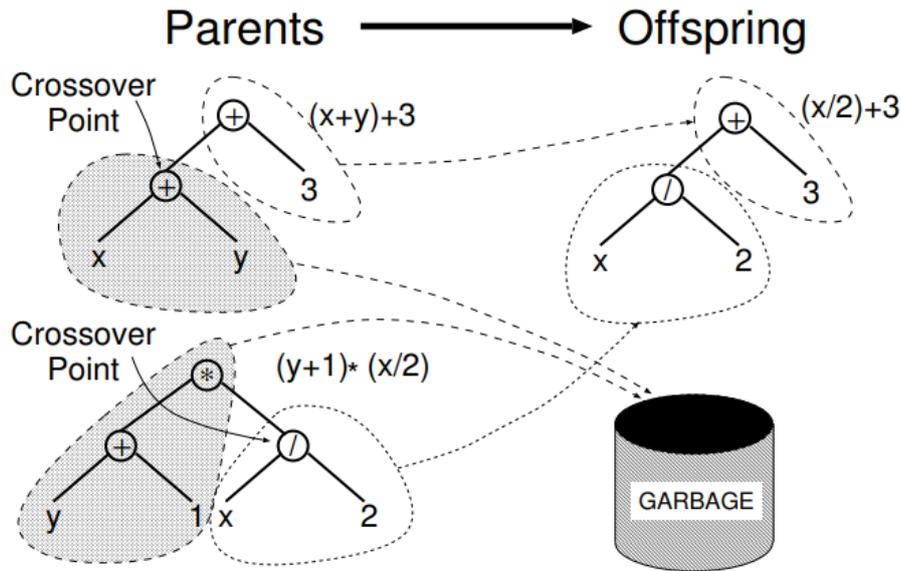


Figura 6.1: Diagrama esquemático que muestra cómo actúa la operación genética *cruce* (crossover) sobre dos árboles (parents) para producir otro completamente distinto (offspring).

evaluar árboles cada vez más grandes se refleja en el tiempo de cómputo para validar el modelo actual. Como la complejidad de las expresiones optimizadas está directamente relacionada a la de los datos a modelar[33], al cabo de cierto tiempo las expresiones pueden llegar a ser tan largas (la longitud de un árbol se define en función de la cantidad de nodos que lo conforman) que se produce una meseta en la optimización llamada *bloat*. Dependiendo de la capacidad de cómputo de la que se disponga para desplegar el algoritmo genético, este estancamiento puede producirse más tarde o más temprano. Por lo que es necesario introducir moderadores de crecimiento que corten aquellas expresiones que son demasiado largas en comparación con otras de menor longitud y *fitness* equivalente[1].

Eureqa es una implementación de RS, la más avanzada en el campo, creada entre 2007 y 2011 por Josh Bongard, Michael Schmidt y Hod Lipson en Cornell University. Desde entonces se ha utilizado en diferentes campos del conocimiento humano con el propósito de develar las relaciones internas en un conjunto de datos sin ningún bias humano y encontrar uno o más modelos subyacentes que los exprese. Para acelerar la búsqueda, esta herramienta tiene soporte para trabajar no sólo con los procesadores de los que el usuario dispone sino también para hacerlo con procesadores en la nube de Amazon Web Services o servidores *in house*.

En esta tesis utilizamos Eureqa como herramienta de RS para modelar el perfil de ondas gravitacionales. Nuestro enfoque es usar un muestreo nuevo para intercambiar recursos computacionales por algoritmos innovativos.

Antes de introducirnos de lleno en ello, hagamos una digresión para comentar un pequeño experimento con Eureqa que deja en relieve su poder.

6.2.1. RS para encontrar leyes de la naturaleza

De mecánica clásica recordemos la 2da ley de Newton:

$$\vec{F} = m\vec{a}. \quad (6.1)$$

Para el caso del péndulo simple, esta ecuación en coordenadas polares es

$$\ddot{\theta} = -k \sin(\theta) \quad k = -g/l, \quad (6.2)$$

donde g es la aceleración gravitatoria y l la longitud del péndulo.

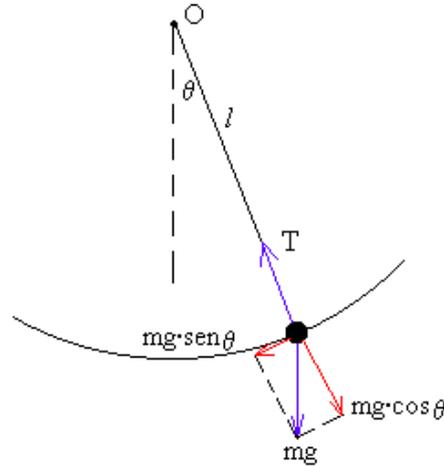


Figura 6.2: Modelo para el péndulo simple: masa puntual m con distancia al punto O fija y sujeta a un campo gravitatorio de magnitud g constante.

Lo que se hizo fue integrar numéricamente esta ecuación con parámetro $k = 0.5$, condición inicial $\theta_0 = \frac{\pi}{2}$ y una grilla temporal equiespaciada. Luego se introdujo los datos numéricos obtenidos para θ vs. t en Eureka y se pidió que encuentre la ley matemática correspondiente en la forma de una ecuación diferencial de tipo

$$\ddot{\theta} = f(\dot{\theta}, \theta, t). \quad (6.3)$$

Obsérvese que esta ecuación es la forma más general de una de segundo orden, de modo que no se está introduciendo ningún bias.

Eureka encontró de forma exacta (en oposición a numérica) la ecuación (6.2) en una computadora portátil, en aproximadamente 1 segundo. Esto es, se halló la segunda ley de Newton, ec. (6.1).

El lector puede repetir este ejercicio introduciendo ruido en los datos -esto es, en adición a los ruidos intrínsecos por haber usado como espacio de entrenamiento soluciones numéricas- o incluso probando distintas condiciones iniciales para el péndulo. La ecuación correcta es hallada nuevamente como propuesta de ley matemática. En el caso de condiciones iniciales correspondientes a ángulos pequeños, la aproximación de oscilador armónico es, como puede esperarse, también sugerida como modelo. En síntesis, **un robot puede encontrar de forma exacta la 2da ley de Newton usando los datos de un péndulo, sin supervisión humana.**

Conectando con otro nodo histórico, esto es exactamente lo mismo que hizo Kepler para encontrar sus leyes fenomenológicas sobre el movimiento planetario, analizando las

observaciones astronómicas de Tycho Brahe. Por ejemplo, supongamos que vivimos en un mundo muy extraño, donde no se conociera la 2da ley de Newton, pero se tuviera un robot que implementara PG; la tarea del hombre en busca de las leyes de la dinámica de los cuerpos materiales residiría únicamente en el paso inductivo del método científico, pues el robot haría todo el trabajo pesado del análisis fenomenológico de los sistemas físicos. El alcance de esta reflexión y el experimento presentado, cuando uno piensa en la cantidad de aplicaciones, no puede ser sobreenfatizado.

Es importante aclarar que, con la implementación de RS, no se halla un único modelo, sino un conjunto de los mismos que representan bien los datos en el dominio de interés, con distinto nivel de precisión y complejidad computacional en sus evaluaciones. En el ejemplo del péndulo, si bien la 2da ley de Newton fue rápidamente hallada en forma exacta, igualmente fueron propuestas otras expresiones que representan al péndulo en el rango temporal de los datos introducidos como input, aunque con mayor error.

Para terminar este capítulo, mencionamos que, con estos métodos, también pueden encontrarse formas implícitas tales como superficies de energía constante o incluso leyes físicas que gobiernan sistemas mecánicos caóticos, tales como el doble péndulo [35].

Capítulo 7

Resultados generales

En este capítulo presentamos los resultados generales de esta tesis, a la vez que repasamos algunos de los puntos desarrollados en el capítulo 3 luego de haber discutido las componentes técnicas a lo largo de los capítulos 4, 5 y 6.

Modelo de Referencia

Simular coalescencia de objetos compactos, como agujeros negros (BHs) o estrellas de neutrones (NSs), resolviendo numéricamente las ecuaciones de Einstein, es computacionalmente costoso, requiriendo de días a meses de supercomputación. En [8] se implementaron las técnicas de reducción de dimensionalidad descritas en los capítulos 4 y 5, obteniendo modelos sustitutos para ondas gravitacionales emitidas en la coalescencia de dos BHs, inicialmente sin espín y en órbita cuasi-circular, en el rango de cociente entre las masas $q = m_1/m_2 \in [1, 10]$ y duración de alrededor de 15 órbitas antes de la fusión (*merger*). Los errores alcanzados en los modelos sustitutos respecto de las simulaciones de relatividad numérica, usadas tanto en el entrenamiento como en la validación, fueron comparables a los errores numéricos de las simulaciones mismas. Esto es, el modelo sustituto resultó indistinguible de soluciones numéricas. Aún más, el código usado para entrenamiento y validación (SpEC [3, 38]) es el más preciso en la comunidad de relatividad numérica. Esta labor llevó a producir un mapa suave como modelo sustituto (MP numérico o modelo de referencia; véase sección 3) a partir del conjunto discreto de ondas gravitacionales que se utilizó para entrenamiento, pudiendo no solo aproximar dicho conjunto y predecir soluciones nuevas, sino introducir un factor de aceleración en su evaluación de 10^{6-8} comparado con las simulaciones numéricas.

7.1. Buscando formas cerradas

Los dos grados de polarización de ondas gravitacionales pueden codificarse en una única función compleja

$$h(t, q) := h_+(t, q) + i h_x(t, q), \quad (7.1)$$

donde i denota la unidad imaginaria. Esto simplifica la estructura de las ondas mediante su descomposición polar, produciendo superficies de fase y amplitud simples en comparación con las superficies asociadas a las componentes h_+ y h_x por separado.

En la sección 3.2, a modo de aplicación directa, presentamos una expresión simbólica de alta precisión para la onda gravitacional correspondiente a un solo valor de q ($q = 2$). El enfoque fue encontrar modelos simbólicos para amplitud y fase, y luego reconstruir los dos grados de polarización de la onda en forma simbólica.

En contraste, hacer RS para cada componente h_+ y h_x -por separado y con q fijo- e intentar hallar un modelo simbólico en el tiempo para dichas componentes, no fue posible en nuestra experiencia (en términos de velocidad de convergencia).

En este trabajo, al momento de buscar expresiones simbólicas para la amplitud de una onda gravitacional, se han probado desde sampleos densos y equiespaciados en el tiempo, hasta sampleos poco densos y adaptados a la estructura -con criterios arbitrarios, tanto en locación de los puntos como en cantidad-, logrando ningún éxito o un éxito parcial en la precisión del modelo simbólico obtenido. El problema, como discutimos en el capítulo 3, reside en que, para la amplitud, RS tiene dificultades para resolver la parte del pico y el *ringdown*. Por otro lado, mientras más puntos de entrenamiento y validación se utilizan (no sólo en Eureqa, sino en general, en cualquier algoritmo de RS), el tiempo computacional crece rápidamente, ya que por cada punto se tiene que validar la generación entera para computar su *fitness*. En respuesta a estos obstáculos, recurrimos al método de RB+EIM para iluminar el camino.

Se produjo un grillado adaptativo en el dominio del tiempo $t \in [-2750, 100]M$ para todas las ondas con $q \in [1, 10]$. A efectos de estandarizar las dimensiones del dominio (t, q) para entrenamiento, se introdujo un factor de escala de 10^{-3} en el tiempo. La razón para hacerlo fue evitar que RS incurra en la proliferación de constantes innecesarias en el proceso evolutivo[1] y que relente la búsqueda. De esta manera, el intervalo para t queda igual a $[-2.75, 0.1]M$. Este grillado es el mismo para todo q en el intervalo considerado y se obtiene utilizando un conjunto de puntos minimal computados vía Método de Interpolación Empírica en [8](véase cap. 5). De esta manera, se reduce considerablemente la cantidad de datos a utilizar en el entrenamiento de RS, sin detrimento de la estructura subyacente a la amplitud y fase asociada a los perfiles de onda $h_+(t, q)$ y $h_x(t, q)$.

La idea fundamental e intuición inicial de este enfoque es que los nodos EIM son los puntos ‘más relevantes’ y los únicos necesarios para reconstruir todos los perfiles temporales, de modo que deberían ser suficientes para entrenar RS. Esta hipótesis resultó validada con la aplicación de los nodos EIM en este trabajo.

7.2. Los nodos EIM en acción

Notamos que un sampleo temporal equiespaciado es cuanto menos ingenuo, además de inefectivo, para capturar las diferentes estructuras del campo que se está discretizando -e.g. el *inspiral-merger-ringdown* de la amplitud. Para optimizar la captura de la estructura relevante de un modelo y a la vez minimizar la cantidad de datos utilizados para ello, una opción es proponer grillas adaptativas. Pero hacer esto *a mano*, con cada problema de interés, no es general ni escalable, y ya notamos que, al menos en nuestra experiencia, incurrir en estas manualidades no arroja resultados precisos.

En los capítulos sobre Reducción Dimensional (4 y 5) describimos un enfoque completamente general y escalable que captura la información relevante de cualquier estructura de datos parametrizados, combinando algoritmos de compresión en las dos dimensiones - la *paramétrica* y la *dual ‘física’* -, para producir un modelo idéntico al original, con un error menor o igual a aquel especificado por el usuario. En la práctica, elegimos este error como el de redondeo de doble precisión, de manera que, a fines prácticos, la compresión es *lossless*.

Con este enfoque se obtienen los puntos ‘más relevantes’ del dominio paramétrico y su dominio dual (el ‘físico’), a saber, los puntos *greedy* y los nodos EIM, para conectarlos

mediante técnicas de fiteo e interpolación y producir un modelo no sólo idéntico al original (que está discretizado), sino también *predictivo*, produciendo información nueva donde antes no la había.

7.3. Resultados

Todos los modelos simbólicos tienen coeficientes de determinación respecto de sus contrapartes de referencia de, al menos, $R^2 \sim 0.999$.

Modelos simbólicos para la amplitud

Sampleamos el tiempo con los nodos EIM y utilizamos una grilla equiespaciada para el parámetro q en el intervalo $[1-10]$. Ya que hay 22 nodos empíricos (igual que el número de elementos de la base reducida) para el modelo original, se produjeron, vía *gwsurrogate*, 22 curvas de referencia para la amplitud -una por nodo temporal, con 90 puntos equiespaciados en q .

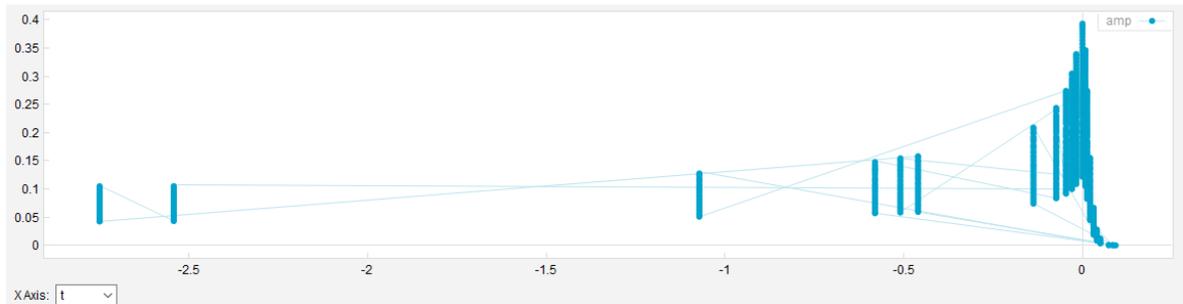


Figura 7.1: Proyección de la amplitud $A(t, q)$ en q . Los nodos EIM resuelven, tal como se esperaría, la zona del *merger* y *ringdown* para todos los valores de q utilizados.

En la fig. 7.1 vemos que los nodos EIM tienden efectivamente a resolver el *merger* y *ringdown*. Por otro lado, la parte del *inspiral* contiene sólo unos pocos nodos, dada la falta de estructura en ese régimen. Esto es, con una cantidad minimal de puntos temporales, EIM captura las ondas para su entrenamiento en RS. El resultado - ya presentado en el capítulo 3 y aquí agregado por completitud - es

$$\begin{aligned}
 A(t, q) = & \{1.37502533181183 \times \exp(\text{atan2}(t, 0.0409895367586908 - t))\} \\
 & / \{3.40043449934568 + q - t - 1.86434379599601 \\
 & \text{gauss}(\text{atan2}(1.1446516014466, q) - 1.49686180948812t) \\
 & \text{gauss}(\text{atan2}(t, 0.0250835926883564 - t \\
 & - 0.108134472792241tq))\} - 0.00178301085458751,
 \end{aligned} \tag{7.2}$$

donde $\text{gauss}(x) := \exp(-x^2)$ y $\text{atan2}(x, y)$ denota la función arcotangente de 2 parámetros.

Modelos simbólicos para la fase

Como se explicó en el capítulo 3, se logró hacer RS para la fase, en todo el intervalo $q \in [1, 10]$, con alta precisión. Sin embargo, encontramos que pequeños errores en la misma se amplifican en el momento de reconstruir las ondas¹.

Por ello realizamos una descomposición del dominio para $q \in [1, 10]$ en 9 subintervalos $[1, 2] - [2, 3] - \dots - [9, 10]$. Se utilizó un muestreo equiespaciado para el tiempo² t (285 puntos) y el parámetro q (20 puntos) y se obtuvieron 9 formas simbólicas para la fase (una por cada subdominio). Las expresiones son:

$$\begin{aligned} \phi_{[1,2]}(t, q) = & 16.9899198245249 * \operatorname{atan2}(0.0307964991839896, t) \\ & * \operatorname{asinh}(8.47135183989294 * t) + 0.517828856434813 * t * \operatorname{cosh}(q) \\ & * \operatorname{asinh}(8.47135183989294 * t) / \operatorname{atan2}(0.0307964991839896, t) \\ & - 155.336099835965 - 23.6826537661638 * t - 1.67469018319234 \\ & * \operatorname{cosh}(q) - 26.3800439393647 * \operatorname{asinh}(t) - 64.7124400428524 \\ & * \operatorname{asinh}(8.47135183989294 * t) \end{aligned} \quad (7.3)$$

$$\begin{aligned} \phi_{[2,3]}(t, q) = & 6.83585963818032 * t^2 + 29.2184323756819 \\ & * t * \exp(t) - 150.134280878147 - 5.63837416811418 * q \\ & - 336.187115348329 * t - 2.03916141015058 * q * t \\ & - 0.462051355197209 * \operatorname{asinh}(224.317310349048 * t) - \\ & 42.1012107484649 * t * \operatorname{asinh}(224.317310349048 * t) \end{aligned} \quad (7.4)$$

$$\begin{aligned} \phi_{[3,4]}(t, q) = & -150.779871664876 - 5.48151705552273 * q \\ & - 20.9244777358138 * t - 40.725476363926 * \operatorname{asinh}(t) \\ & - 1.98410145646586 * q * t - 58.0238952257731 \\ & * \operatorname{atan}(4.54076522147913 * t) - 27.5236418372851 \\ & * \operatorname{atan}(27.0569353214333 * t) * \operatorname{atan}(4.54076522147913 * t) \end{aligned} \quad (7.5)$$

$$\begin{aligned} \phi_{[4,5]}(t, q) = & 8.5721194612964 * \operatorname{tanh}(t) + 4.77594286022371 * t^2 \\ & + 1.71392847899927 * q * t * \operatorname{tanh}(t) - 153.735830018958 \\ & - 4.65044447419562 * q - 301.17672677624 * t - 0.401467421747743 \\ & * \operatorname{asinh}(300.050625169173 * t) - 34.6087658515619 \\ & * t * \operatorname{asinh}(300.050625169173 * t) \end{aligned} \quad (7.6)$$

$$\begin{aligned} \phi_{[5,6]}(t, q) = & 34.9967402431729 * \operatorname{atan2}(2.88396955316236, \\ & \exp(18.5550015973867 * t + (18.5550015973867 * t^2)^{0.56242721246326})) \\ & - 144.334463413931 - 4.56653604042429 * q - 33.8215169236648 * t \\ & - 53.5938957492291 * \exp(t) - 1.05498674575652 * q * t \\ & * \operatorname{atan2}(0.0457688186271425, \exp(18.5550015973867 * t \\ & + (18.5550015973867 * t^2)^{0.56242721246326})) \end{aligned} \quad (7.7)$$

¹Para detalles ver el Apéndice A.

²La simpleza estructural de la fase permite que RS converja rápidamente para cualquier distribución de puntos.

$$\begin{aligned}
\phi_{[6,7]}(t, q) = & 1.16459728558136 * t^2 + 2.04818853464955 \\
& * \operatorname{asinh}(-3031.15606950987 * t) + 5.63150899599863 \\
& * \operatorname{atan2}(5.54505004946533, -3031.15606950987 * t) \\
& + 19.5350382862564 * t * \operatorname{asinh}(-3031.15606950987 * t) \\
& - 164.847087315508 - 4.24649927087975 * q \\
& - 239.435021215528 * t - 1.54209668558109 * q * t
\end{aligned} \tag{7.8}$$

$$\begin{aligned}
\phi_{[7,8]}(t, q) = & 42.5234029423116 * \operatorname{atan2}(-0.454286929738614, t) \\
& - 92.557645930759 - 3.87693080235069 * q \\
& - 21.6846366746129 * t - 1.40441380466267 * t * q \\
& - 103.004616315544 * \operatorname{asinh}(3.4964623078196 * t) \\
& - 31.6037083633658 * \operatorname{atan}(19.0682642359228 * t) \\
& * \operatorname{asinh}(3.4964623078196 * t)
\end{aligned} \tag{7.9}$$

$$\begin{aligned}
\phi_{[8,9]}(t, q) = & 0.223595449207837 * t^2 + 0.0228895534195837 \\
& * \exp(53.3803388021468 * t) + 108.64303226437 * t \\
& * \operatorname{asinh}(\operatorname{asinh}(-24.5987724166689 * t)) - 162.16004867221 \\
& - 3.50199154150564 * q - 307.395374115204 * t \\
& - 0.565057456771371 * t * q \\
& * \operatorname{asinh}(\operatorname{asinh}(-24.5987724166689 * t))
\end{aligned} \tag{7.10}$$

$$\begin{aligned}
\phi_{[9,10]}(t, q) = & 10.6554882562835 * \operatorname{atan2}(t, 0.561000822409326) \\
& + 0.390622102213681 * q * t^2 - 164.868219149625 \\
& - 3.16971455362515 * q - 293.785912743941 * t \\
& - 33.3325708174699 * t * \operatorname{asinh}(174.055527284276 * t) \\
& - 0.00463931687452369 * q^2 * \operatorname{asinh}(174.055527284276 * t)
\end{aligned} \tag{7.11}$$

7.4. Validación

Hallados los modelos simbólicos para amplitud y fase, la intención última de este trabajo es poder reconstruir los dos grados de polarización de las ondas h_+ y h_x . Para saber qué tan bueno es el modelo simbólico en dicha tarea, debemos cuantificar la fidelidad del mismo para las ondas respecto de sus contrapartes de referencia. A simple vista, las ondas coinciden en la superposición incluso para la peor coincidencia (*matching*) en el intervalo $[1, 10]$ para q (véase fig. 7.2). Con el objeto de cuantificar esta observación, calculamos el overlap S entre las ondas simbólicas y las de referencia para una grilla densa de puntos q en el dominio $[1, 10]$, usando 60 puntos equiespaciados -que implica usar ondas no presentes en el espacio de entrenamiento- por cada intervalo $[1, 2] - [2, 3] - \dots - [9, 10]$. La integración se realizó en todo el dominio temporal con la resolución original (28501 ptos. temporales en el intervalo $[-2750, 100]$ M). Siendo más precisos, se calculó

$$S(h_{+o}(q), h_{+s}^i(q)) \quad i = 1, 2, \dots, 9, \tag{7.12}$$

para los 9 modelos simbólicos encontrados en cada intervalo. En la fig. 7.3 se muestra la dependencia en q del overlap S para el modo h_+ ; los resultados para h_x son similares. Como puede observarse, el overlap en toda la región $q \in [1, 10]$ es alto, lo suficiente para fines observacionales como LIGO y VIRGO.

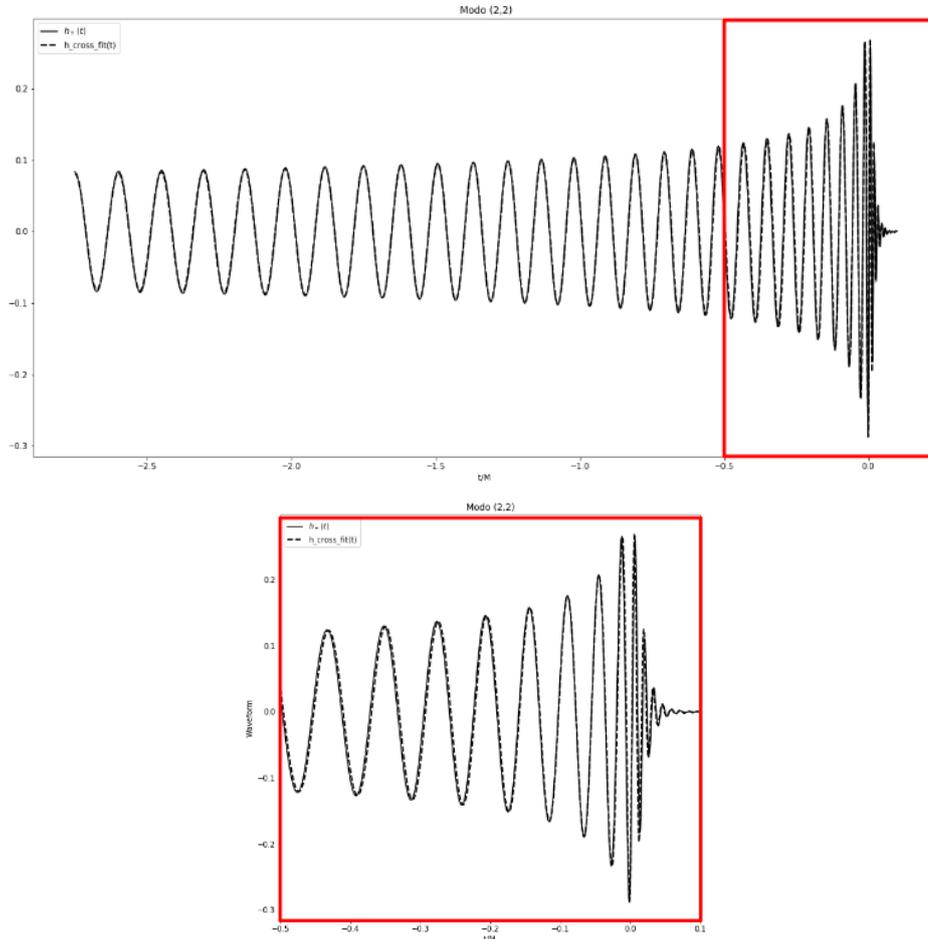


Figura 7.2: Onda simbólica $h_+(t, q = 2, 97)$ en superposición con su correspondiente onda de referencia, para la peor coincidencia (mínimo overlap $S = 0.9905$). Marco rojo: *zoom in* al merger.

Con esto finalizamos la presentación de los resultados principales de esta tesis.

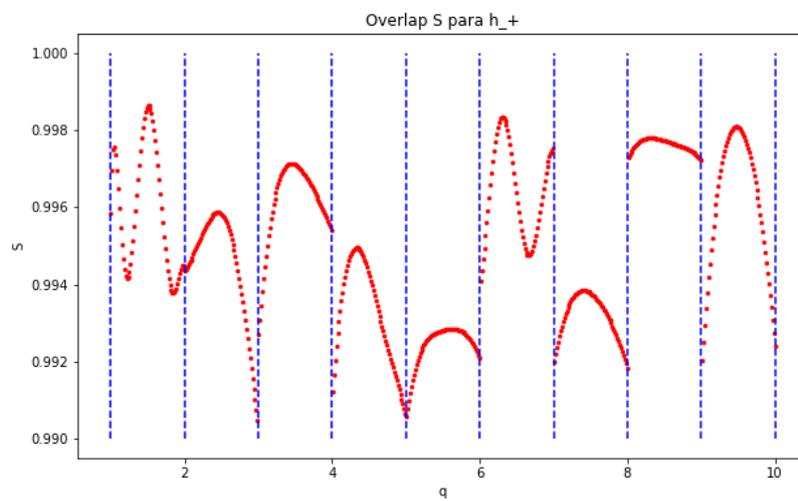


Figura 7.3: Overlaps $S(q)$ vs q para el modo $h_+(t, q)$. $\min(S)=0.9905$. $\max(S)=0.9986$. Las líneas punteadas azules delimitan cada subdominio.

Capítulo 8

Conclusión

Hemos hallado expresiones simbólicas para ondas gravitacionales correspondiente al modo multipolar (2,2), emitidas por la colisión de dos agujeros negros, para órbita cuasi-circular e inicialmente sin espín. Nos restringimos al dominio de cociente entre las masas igual a $[1, 10]$ y rango temporal correspondiente a $[-2750, 100]M$, donde M es la masa total (ADM) del espaciotiempo. La construcción del modelo simbólico siguió una metodología *ab-initio*, es decir, no se incurrió en ninguna simplificación de las ecuaciones de Einstein. La precisión mínima obtenida por el modelo corresponde al valor 0.9905 de overlap o, equivalentemente, a un error global máximo del 1% que, a efectos prácticos, no presenta una pérdida de resolución respecto de simulaciones de relatividad numérica. En comparación con simulaciones de supercomputadoras de las ecuaciones de Einstein, el modelo puede evaluarse en una computadora de escritorio en menos de 1 seg.

Cabe notar que el enfoque presentado -que combina en sinergia bases reducidas, el método de interpolación empírica, modelos predictivos y programación genética- es, por diseño, abstracto y aplicable a un número ilimitado de campos dentro de ciencia de datos.

Apéndice A

Errores en la fase de ondas gravitacionales

En este apéndice presentamos un modelo simbólico para la fase en todo el dominio de $q \in [1, 10]$ y discutimos cómo, aunque el modelo es de alta precisión, pequeños errores en la fase se amplifican en las ondas gravitacionales en el momento de su reconstrucción.

Reconstrucción

EL modelo simbólico para todo el dominio de q corresponde al presentado en las ecs. 3.6 y 3.7. Elegimos un valor $q \in [1, 10]$ arbitrario para reconstruir en el tiempo el modo h_+ correspondiente y así poder compararlo, por superposición, con el modo de referencia. En la fig. A.1 puede notarse la distorsión en la fase en la onda simbólica reconstruida. Encontramos que esta sensibilidad está presente en todos los casos (valores de q), lo que nos llevó a mejorar la precisión del modelo para poder reconstruir exitosamente una onda para q arbitrario. En la fig. A.2 se aprecia mejor la pérdida de fidelidad en la reconstrucción de las ondas: mostramos la dependencia de la función $S(q)$ con q .

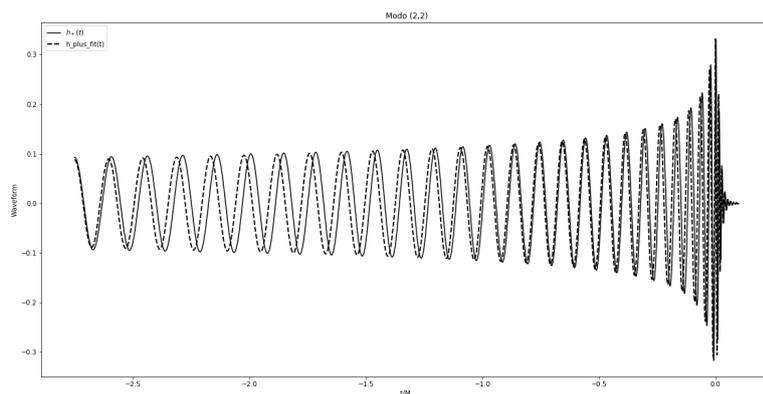


Figura A.1: Reconstrucción del modo h_+ para un q genérico.

Puede verse cómo falla el modelo en reproducir la onda original, haciendo necesario recurrir a particiones del intervalo $q \in [1, 10]$ para mejorar la precisión.

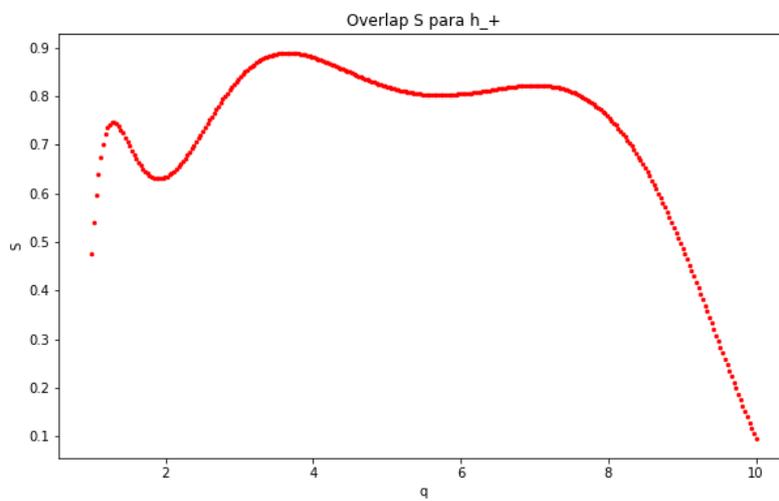


Figura A.2: Overlap $S(q)$ vs q para el modelo simbólico $h_{+s}(t, q)$. $\min(S)=0.0944$. $\max(S)=0.8900$.

Apéndice B

Métodos de regresión simbólica

Si bien Eureka data del 2007-2011, es aún superior a otros métodos *open source* de PG y RS modernos. Por ejemplo: entrenamos con el paquete open source *gplearn* de python para RS[1] la superficie correspondiente a la fase $\phi(t, q)$ y se obtuvo lo que puede verse en la figura B.1. Los resultados dejan mucho que desear. Se invirtieron

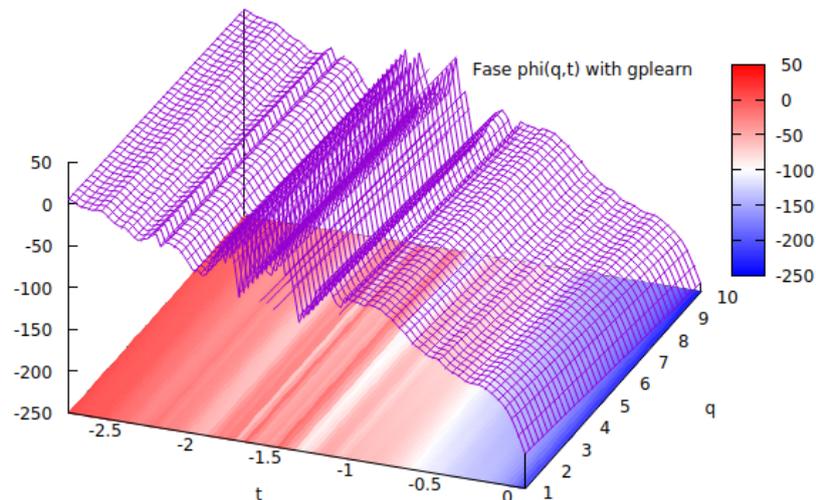


Figura B.1: Expresión simbólica de la fase $\phi(t, q)$ con el paquete *gplearn* de RS de python.

alrededor de 30 minutos de entrenamiento ya que, con los parámetros especificados (parsimonia, población, fitness, etc.), el error absoluto medio no decreció de forma significativa empleando más tiempo en las diferentes pruebas realizadas.

El diccionario del paquete *gplearn* es limitado (funciones aritméticas básicas, trigonométricas, etc.), así también el de sus métricas para evaluar la función *fitness*. Eureka es poderoso debido a su amplio diccionario y su extensa lista de métricas. Presenta un sofisticado método de validación que previene *overfittings*. En contraposición, la ventaja de un paquete *open source* como *gplearn*, o similares, es el de poder expandirlo.

Bibliografía

- [1] gplearn. <https://gplearn.readthedocs.io/en/stable/#>.
- [2] gwsurrogate 0.9.6. <https://pypi.org/project/gwsurrogate/>.
- [3] Spectral einstein code. <https://www.black-holes.org/code/SpEC.html>.
- [4] ANTIL, H., FIELD, S. E., HERRMANN, F., NOCHETTO, R. H., AND TIGLIO, M. Two-step greedy algorithm for reduced order quadratures. *Journal of Scientific Computing* 57, 3 (Dec 2013), 604–637.
- [5] BAKER, J. G., CENTRELLA, J., CHOI, D.-I., KOPPITZ, M., AND VAN METER, J. *Phys. Rev. Lett.* 96 (2006), 111102.
- [6] BARRAULT, M., MADAY, Y., NGUYEN, N. C., AND PATERA, A. T. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique* 339, 9 (2004), 667 – 672.
- [7] BINEV, P., COHEN, A., DAHMEN, W., DEVORE, R. A., PETROVA, G., AND WOJTASZCZYK, P. Convergence rates for greedy algorithms in reduced basis methods. *SIAM J. Math. Analysis* 43, 3 (2011), 1457–1472.
- [8] BLACKMAN, J., FIELD, S. E., GALLEY, C. R., SZILÁGYI, B., SCHEEL, M. A., TIGLIO, M., AND HEMBERGER, D. A. Fast and accurate prediction of numerical relativity waveforms from binary black hole coalescences using surrogate models. *Phys. Rev. Lett.* 115 (Sep 2015), 121102.
- [9] CAMPANELLI, M., LOUSTO, C. O., MARRONETTI, P., AND ZLOCHOWER, Y. *Phys. Rev. Lett.* 96 (2006), 111101.
- [10] CANIZARES, P., FIELD, S. E., GAIR, J., RAYMOND, V., SMITH, R., ET AL. Accelerated gravitational-wave parameter estimation with reduced order modeling.
- [11] CANIZARES, P., FIELD, S. E., GAIR, J. R., AND TIGLIO, M. Gravitational wave parameter estimation with compressed likelihood evaluations. *Phys. Rev. D* 87 (2013), 124005.
- [12] CAUDILL, S., FIELD, S. E., GALLEY, C. R., HERRMANN, F., AND TIGLIO, M. Reduced basis representations of multi-mode black hole ringdown gravitational waves. *Classical and Quantum Gravity* 29, 9 (apr 2012), 095016.

- [13] CAUDILL, SARAH AND FIELD, SCOTT E. AND GALLEY, CHAD R. AND HERRMANN, FRANK AND TIGLIO, MANUEL. Reduced basis representations of multi-mode black hole ringdown gravitational waves, 2011. submitted to LSC review.
- [14] CHATURANTABUT, S., AND C. SORENSEN, D. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Scientific Computing* 32 (01 2010), 2737–2764.
- [15] D BROWN ET AL. *J. Opt.* 18 025604, 2016.
- [16] EFTANG, J. L., HUYNH, D. B., KNEZEVIC, D. J., AND PATERA, A. T. A two-step certified reduced basis method. *J. Sci. Comput.* 51, 1 (Apr. 2012), 28–58.
- [17] EFTANG, J. L., AND STAMM, B. Parameter multi-domain empirical interpolation. *International Journal for Numerical Methods in Engineering* 90, 4 (2012), 412–428.
- [18] EFTANG, J. L., AND STAMM, B. Parameter multi-domain empirical interpolation. *International Journal for Numerical Methods in Engineering* 90, 4 (2012), 412–428.
- [19] FARES, M., HESTHAVEN, J., MADAY, Y., AND STAMM, B. The reduced basis method for the electric field integral equation. *Journal of Computational Physics* 230, 14 (2011), 5532 – 5555.
- [20] FIELD, S. E., GALLEY, C. R., HERRMANN, F., HESTHAVEN, J. S., OCHSNER, E., AND TIGLIO, M. Reduced basis catalogs for gravitational wave templates. *Phys. Rev. Lett.* 106 (2011), 221102.
- [21] FIELD, S. E., GALLEY, C. R., HESTHAVEN, J. S., KAYE, J., AND TIGLIO, M. Fast prediction and evaluation of gravitational waveforms using surrogate models. *Phys. Rev. X* 4 (Jul 2014), 031006.
- [22] FIELD, S. E., GALLEY, C. R., AND OCHSNER, E. Towards beating the curse of dimensionality for gravitational waves using Reduced Basis. *Phys. Rev. D* 86 (2012), 084046.
- [23] GAUTSCHI, W. The condition of vandermonde-like matrices involving orthogonal polynomials. *Linear Algebra and its Applications* 52-53 (1983), 293 – 300.
- [24] HOLST, M., SARBACH, O., TIGLIO, M., AND VALLISNERI, M. The Emergence of Gravitational Wave Science: 100 Years of Development of Mathematical Theory, Detectors, Numerical Algorithms, and Data Analysis Tools. *Bull. Am. Math. Soc.* 53, 4 (2016), 513–554.
- [25] KOZA, J. R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.
- [26] MADAY, Y., NGUYEN, N. C., PATERA, A. T., AND PAU, S. H. A general multipurpose interpolation procedure: the magic points. *Communications on Pure and Applied Analysis* 8 (2009), 383–404.

- [27] MAGGIORE, M. *Gravitational waves: Volume 1: Theory and experiments*, 1st ed. Oxford University Press, 2008.
- [28] MAGGIORE, M. *Gravitational Waves: Volume 2: Astrophysics and Cosmology*. Oxford University Press, 2018.
- [29] PETER BENNER, ALBERT COHEN, M. O., AND WILLCOX, K. *Model Reduction and Approximation: Theory and Algorithms*. SIAM.
- [30] PINKUS, A. *N-Widths in Approximation Theory (ERGEBNISSE DER MATHEMATIK UND IHRER GRENZGEBIETE 3 FOLGE)*. Springer Verlag, 1985.
- [31] PRETORIUS, F. *Phys. Rev. Lett.* 95 (2005), 121101.
- [32] RICCARDO POLI, NICHOLAS F. MCPHEE, W. B. L. *A Field Guide to Genetic Programming*. Lulu.com, London, United Kingdom, 1992.
- [33] RYLANDER, B., SOULE, T., AND FOSTER, J. Computational complexity, genetic programming, and implications. vol. 2038, pp. 348–360.
- [34] SARBACH, O., AND TIGLIO, M. Continuum and discrete initial-boundary value problems and einstein’s field equations. *Living Reviews in Relativity* 15, 1 (Aug 2012), 9.
- [35] SCHMIDT, M., AND LIPSON, H. Distilling free-form natural laws from experimental data. *Science* 324, 5923 (2009), 81–85.
- [36] SCHMIDT, M., AND LIPSON, H. *Symbolic Regression of Implicit Equations*. Springer US, Boston, MA, 2010, pp. 73–85.
- [37] SHANNON, C., GALLAGER, R., AND BERLEKAMP, E. Lower bounds to error probability for coding on discrete memoryless channels. ii. *Information and Control* 10, 5 (1967), 522 – 552.
- [38] SZILÁGYI, B. Key elements of robustness in binary black hole evolutions using spectral methods. *International Journal of Modern Physics D* 23 (2014).
- [39] TIGLIO, M., AND VILLANUEVA, A. *Afternotes on Reduced Basis*. Cambridge University Press. Unpublished.