



GRADO EN MATEMÁTICA COMPUTACIONAL

ESTANCIA EN PRÁCTICAS Y PROYECTO FINAL DE GRADO

---

**Estudio de las curvas geodésicas y los  
tensores curvatura de Riemann y Ricci en  
variedades semiriemannianas**

---

*Autor:*  
Fernán GONZÁLEZ IBÁÑEZ

*Supervisor:*  
Juan F. DORADO SÁNCHEZ  
*Tutor académico:*  
Vicent GIMENO GARCÍA

Fecha de lectura: 22 de julio de 2021  
Curso académico 2020/2021



## Resumen

En este trabajo partimos de los conocimientos adquiridos en la asignatura de Geometría diferencial y topología para realizar un estudio de las curvas geodésicas y la curvatura de ciertas variedades. Comenzaremos introduciendo la generalización del concepto de superficie que se vio en dicha asignatura a dimensiones superiores, variedades, para posteriormente centrarnos en las variedades diferenciables difeomorfas a  $\mathbb{R}^4$  e introduciremos el concepto de tensor métrico para poder estudiar las propiedades de estas, enfocándonos en las curvas geodésicas y el tensor curvatura.

Para poder realizar un estudio más detallado se desarrollaron una serie de funciones de Python para calcular automáticamente los símbolos de Christoffel, las ecuaciones de las curvas geodésicas y los tensores de curvatura, así como la curvatura escalar. Finalmente, las pondremos en práctica estudiando diferentes métricas: la métrica de Poincaré, la métrica de Minkowski y la métrica de Schwarzschild.

## Palabras clave

Geometría diferencial, Geodesicas, Curvatura, Símbolos de Christoffel

## Keywords

Differential Geometry, Geodesic curve, Curvature, Christoffel symbols



# Índice general

<b>1. Introducción</b>	<b>9</b>
1.1. Contexto y motivación del proyecto . . . . .	9
1.2. Estructura de la memoria . . . . .	9
<b>2. Estancia en prácticas</b>	<b>11</b>
2.1. Introducción . . . . .	11
2.2. Objetivos del proyecto formativo . . . . .	12
2.3. Explicación detallada del proyecto realizado en la empresa . . . . .	12
2.3.1. Conceptos previos . . . . .	12
2.3.2. Etapas del desarrollo de las prácticas . . . . .	14
2.4. Conclusiones y valoración personal . . . . .	17
<b>3. Memoria TFG</b>	<b>19</b>
3.1. Introducción . . . . .	19
3.2. Conceptos previos . . . . .	19
3.2.1. Conceptos básicos de álgebra lineal y multilineal . . . . .	20

3.2.2.	Conceptos básicos de topología . . . . .	21
3.3.	Variedades diferenciables . . . . .	23
3.4.	El espacio euclídeo $\mathbb{R}^n$ como variedad diferenciable . . . . .	31
3.5.	Tensor Métrico . . . . .	34
3.5.1.	Ejemplos de variedades riemannianas y semi-riemannianas . . . . .	37
3.6.	Conexiones . . . . .	39
3.6.1.	Conexión de Levi-Civita . . . . .	41
3.7.	Curvatura . . . . .	46
<b>4.</b>	<b>Resultados</b>	<b>53</b>
4.1.	Códigos . . . . .	53
4.1.1.	Código para calcular las geodésicas . . . . .	54
4.1.2.	Código para calcular las curvaturas . . . . .	56
4.1.3.	Ejemplo de la ejecución de las funciones . . . . .	61
4.2.	Estudio de las curvas geodésicas y la curvatura en diferentes métricas . . . . .	66
4.2.1.	Métrica de Poincaré . . . . .	66
4.2.2.	Métrica de Minkowski . . . . .	71
4.2.3.	Métrica de Schwarzschild . . . . .	76
<b>5.</b>	<b>Conclusiones</b>	<b>93</b>
<b>A.</b>	<b>Código Nomograma</b>	<b>97</b>
<b>B.</b>	<b>Estructuras maximales</b>	<b>111</b>







# Capítulo 1

## Introducción

### 1.1. Contexto y motivación del proyecto

Este proyecto abarca la realización del Trabajo de Fin de Grado en el Grado de Matemática Computacional. Tiene el objetivo de introducción a la geometría diferencial mediante el análisis de ciertas métricas, así como la presentación de la memoria de prácticas de realizadas en la empresa PeRTICA Análisis Estadísticos S.L..

El año pasado durante el Cicle de Conferències Ciutat de Castelló hubo una conferencia con el título "Relatividad general, agujeros negros y radiación de Hawking" impartida por Víctor Aldaya. Aunque fue una conferencia bastante atípica sirvió como punto de partida para mi curiosidad sobre el tema de la Relatividad General y sus bases geométricas. Por eso decidimos realizar este trabajo final de grado.

### 1.2. Estructura de la memoria

Esta memoria esta dividida en grandes dos capítulos con sus correspondientes secciones: El primero corresponde a la estancia en prácticas en la empresa PeRTICA Análisis Estadísticos S.L.. En ella se explicará la estructura de la empresa, las habilidades y conocimientos adquiridos y los proyectos en los que he participado.

En la segunda parte del trabajo se realizará una introducción a las variedades diferenciables y las métricas semi-riemannianas y riemannianas, y se estudiarán las curvas geodésicas y la curvatura de ciertas variedades.



## Capítulo 2

# Estancia en prácticas

### 2.1. Introducción

La estancia en prácticas se realizó entre los meses de noviembre de 2020 y febrero de 2021 en la empresa PeRTICA Análisis Estadísticos S.L. en la localidad madrileña de Getafe. Esta empresa cuenta con más de 10 años de experiencia en el análisis estadístico de datos y soluciones tecnológicas en torno al dato. PeRTICA es una consultora que usa como lenguaje de programación principal SAS y cuenta con más de una veintena de empleados distribuidos por distintas zonas del mundo con sede en Getafe. Está compuesta de las siguientes áreas:

- **Busines Analytics.** El objetivo de esta parte de la empresa es obtener valor de los datos del negocio, por medio de implementación de las soluciones más ajustadas a las necesidades del cliente, en las diferentes áreas de la empresa. Dan soluciones basadas en torno al dato para distintos sectores como pueden ser: manufactura y logística, análisis de seguros, análisis de telecomunicaciones ... [10]
- **Life Scinces.** Aplican los conocimientos en estadística, informática y medicina para aportar el mayor valor posible a los objetivos que tiene el cliente, ya sea un laboratorio farmacéutico o una unidad de investigación. Son expertos en realizar investigación clínica y epidemiología y analítica en *Life Science*. [10]

La principal motivación de esta estancia en prácticas es poner en práctica las diferentes herramientas que se nos han enseñado durante la carrera al ámbito laboral y emplearlas en el campo de *data analytics*, desde el procesamiento inicial de los datos hasta la generación de modelos predictivos de estos.

Durante mi estancia en práctica formé parte de las diferentes ramas que tiene la empresa Bussines Analytics, bajo la supervisión Ana Gentil, y de Life Sciences, bajo la supervisión de de Juan F. Dorado y Carlos Goetz.

## 2.2. Objetivos del proyecto formativo

El objetivo principal de mi proyecto formativo fue la introducción al campo del *data analysis* basándonos en los conocimientos previos adquiridos a lo largo de la carrera de programación y estadística. Para ello se me inició en el paquete de software estadístico SAS, con este objetivo realicé los cursos de SAS Programming 1 y 2. Posteriormente para formarme en el análisis estadístico de datos para ello realicé al cursos de SAS SQL 1. Para complementar mi formación participé en dos proyectos de análisis de datos médicos en el departamento de Life Science y culminé mi estancia en prácticas participando en el proyecto de la creación de nomogramas automáticos mediante SAS.

## 2.3. Explicación detallada del proyecto realizado en la empresa

### 2.3.1. Conceptos previos

En esta sección se van a introducir diferentes conceptos de los que vamos a hablar durante la explicación del proyecto, como pueden ser las librerías y software usado, además de conceptos como los nomogramas.

**SAS** o *Statistical Analysis System* es una suite de software estadístico desarrollado por el SAS Institute para el manejo y análisis de datos, además de para Business Intelligence y Data Mining. Esta *suite* es la usada por la empresa, principalmente para proporcionar las soluciones a sus clientes. Usé esta suite para realizar el procesamiento y análisis de los datos y para la aplicación de algoritmos de clasificación supervisada.

El lenguaje de programación SAS está diseñado principalmente para estadística, pero tiene muchos programas para las diferente áreas del mundo laboral, como puede ser SAS Miner, para la minería de Datos o SAS QUALITY ANALYTIC SUITE, para el análisis de calidad. Además, este lenguaje presenta algunas características muy llamativas, como puede ser la capacidad de realizar consultas sobre las tablas con código SQL. Los principales programas de la suite de SAS que he usado durante mi estancia en prácticas han sido: SAS 9.4, SAS Studio, SAS Enterprise Guide y SAS Viya.

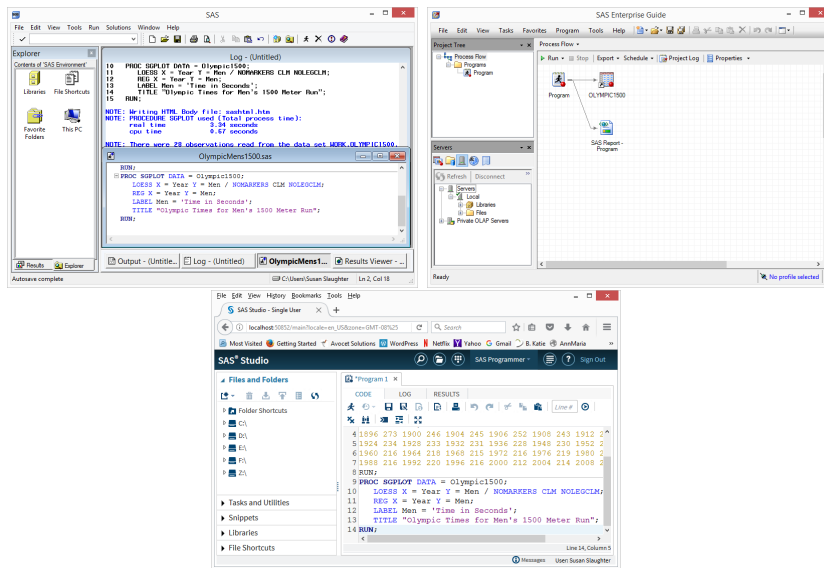


Figura 2.1: Captura de pantalla de los diferentes programas usados durante mis prácticas. De izquierda a derecha y de arriba a abajo: SAS 9.4, SAS Enterprise Guide y SAS Studio

En el inicio de las prácticas comencé usando SAS Studio, ya que es el intérprete de SAS más amigable con el usuario y es el que se usaba en el curso, aunque no el más extendido por la empresa. Este programa nos permite realizar un tratamiento automático de los datos y realizar el análisis simplemente mediante la interfaz de usuario.

Posteriormente, me enseñaron a usar SAS Enterprise Guide ya que era el que se usaba más en la empresa. Es un programa muy similar a SAS Studio, que te permite acceder al código generado y poder modificarlo. Aunque no tanto como SAS 9.4, este programa es el que usé durante la mayoría de mi estancia en prácticas debido a que es el programa más configurable y con el que más fácil era operar con las macros en la empresa. A pesar de no ser un programa tan *user friendly* y no poder diseñar mediante la interfaz de usuario, el *pipeline*, para procesamiento de los datos se podía hacer escribiendo código. De esta forma fue como más aprendí de SAS y su funcionamiento.

Y finalmente SAS Viya, que fue usado al final de mi estancia en prácticas, ya que la librería que se usaría para crear el nomograma estaba hecha para interactuar con SASViya, este software tiene una infinidad de posibilidades.

Los **nomogramas** son unas representaciones gráficas que permiten realizar con rapidez cálculos numéricos aproximados. Existen muchos tipos de nomogramas pero en mi caso el que usé fue un nomograma logístico. Esta clase de nomogramas son usados para poder realizar los

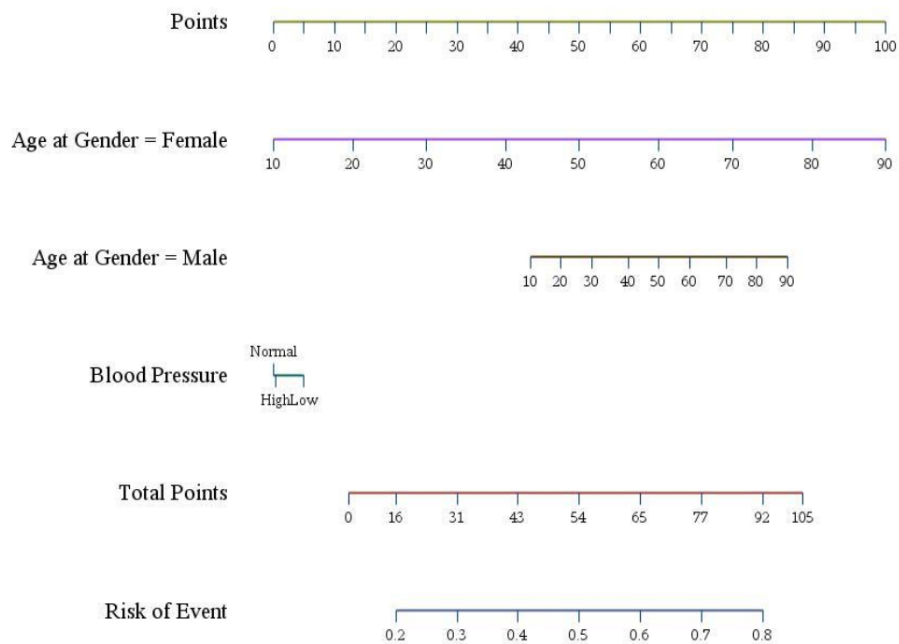


Figura 2.2: Imagen de un nomograma de ejemplo realizado con SAS.

cálculos de una regresión logística de manera aproximada, como el que se puede ver en la figura 2.2. Actualmente los nomogramas son muy usados en el ámbito médico, para representar los riesgos de los pacientes a sufrir un cierto suceso, mostrándoles como actúa cada característica que presenta. Esto ayuda a que los pacientes asimilen mejor la información y no simplemente un número sino un proceso.

Además, para la realización del nomograma se usó el lenguaje de programación SAS y Python las librerías Pandas, para el manejo de las tablas en Python, y SASpy, que permite la interconexión entre un servidor SAS y Python pudiendo así correr código de SAS en Python y obteniendo una sinergia casi perfecta entre las posibilidades que tiene SAS y la simplicidad de Python.

### 2.3.2. Etapas del desarrollo de las prácticas

En el inicio de las prácticas se me asignó al grupo dirigido por Ana Gentil, Bussines Intelligence, al que asistía a las reuniones de grupo y ayudaba en diferentes tareas a los miembros de este. Durante este primer mes estuve realizando los cursos de SAS Programming 1 y 2 y el

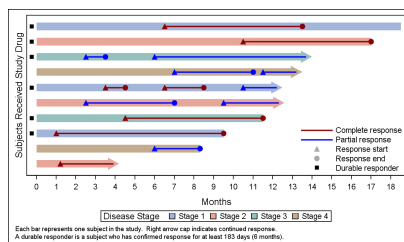


Figura 2.3: Imagen de un swimmer plot de ejemplo realizado con SAS.

curso de IT de la empresa impartido por Felix Vidondo. Durante este periodo se me encargó la realización de un análisis estadístico para la doctora Infante de la unidad de Hematología del Hospital Universitario Infanta Leonor. Este proyecto sería la guía durante toda mi estancia en prácticas. El primer paso en cualquier proyecto de ciencia de datos es el tratamiento de los datos, consiste en la clasificación de las variables, detección de nulos y outliers, así como en detectar posibles fallos en la carga de los datos, asignación de etiquetas identificativas de cada variable . . . Una vez se realizó el preprocesamiento de los datos y el primer análisis exploratorio realicé un informe con los análisis de los datos para la doctora Infante.

En este momento propuse a Carlos Goetz la realización de una macro que sacara el texto en  $\text{\LaTeX}$  en vez de en Word, que nos ahorraría una cantidad significativa tiempo a la hora de realizar los informes, debido a que en Word hacer la numeración de las imágenes debe de hacerse a mano además de poner el color de los textos a mano.

Durante mi segundo mes en la empresa se me introdujo en el proyecto de los doctores Blanca y Somoza del área de alergología del Hospital Universitario Infanta Leonor. Consistió en hacer un informe del análisis estadístico de los datos que nos había facilitado y posteriormente encontrar las mejores variables para realizar una regresión logística para predecir si un paciente sería hospitalizado o no. La primera parte del trabajo fue importar los datos y dar formatos a las variables y posteriormente realizar el análisis exploratorio y la elaboración del informe. Durante la realización de este proceso se me introdujo al programa de recogida de datos médicos Red Cap. Este software sirve para el diseño de formularios para la recogida de datos en estudios médicos. Una vez que se le entregó el informe del análisis estadístico a la doctora Infante tuvimos una reunión con ella y nos encargó el cálculo la incidencia sobre los pacientes que necesitaron hospitalización y hacer un gráfico *Swimmer plot*, como el que se puede observar en la figura 2.3, clasificando los pacientes según su tipo de infección y mostrar la duración del episodio infeccioso.

En el tercer mes de prácticas, debido a las fiestas de navidad y la tormenta Filomena que azotó la península ibérica, las prácticas fueron en formato online y posteriormente siguieron así debido a que tuve los exámenes finales. Durante este mes, los días antes de Navidad estuve

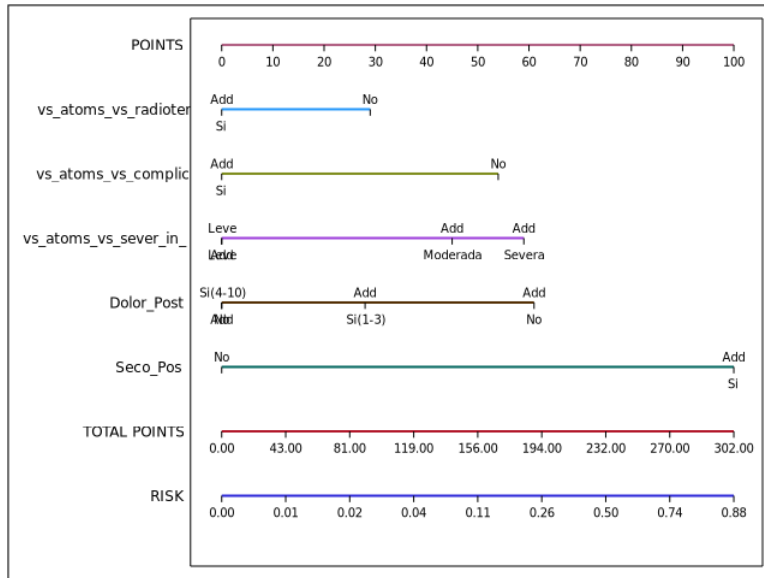


Figura 2.4: Imagen de un nomograma de ejemplo realizado con SASpy.

ayudando en el área de Bussines Intelligence con un proyecto que tenían y necesitaban ayuda. Posteriormente, se nos encargó en el grupo de Life Science la realización de un nomograma logístico automático siguiendo como esquema el artículo escrito por Dongsheng Yang. Este cambio en mi proyecto formativo fue debido a que muchos médicos les estaban pidiendo una forma visual de enseñar a sus pacientes porqué tienen un cierto riesgo y esta es la mejor forma de poder enseñarlo y la realización de un nomograma es muy larga, entonces se intentó automatizar este proceso. Al mismo tiempo continué con mi formación realizando el curso de SQL y de Redes Neuronales en SAS. [6]

Durante el último mes continué con la realización de la creación de nomogramas automáticos mediante SAS, pero, al ver que no podía avanzar utilizando el lenguaje de programación de la empresa comencé a utilizar Python con la librería SASPy. Conseguir así un buena sinergia entre la flexibilidad de de Python y la potencia en cálculo estadístico de SAS. Finalmente, logré alcanzar el objetivo de la realización de un nomograma automático, un ejemplo de la salida de este programa se puede observar en la figura 2.4, y el código se puede encontrar en el Anexo A. Durante este ultimo mes escribí dos artículos a la "wiki" de la empresa, wikiPeRTICA. En ellos expliqué como se debía instalar SASpy y como usar sus diferentes funciones y como utilizar el programa había hecho.



## 2.4. Conclusiones y valoración personal

Mi estancia en practicas en la empresa PeRTICA fue muy satisfactoria debido a que pude aprender un nuevo lenguaje de programación, SAS, y pude aplicar los conocimientos que he aprendido sobre estadística y ampliarlos, así como mejorar mi conocimiento de Python y SQL. Además he podido cimentar una buena base de conocimientos sobre análisis de datos debido a que el equipo que hay en empresa tiene una gran capacidad de transmitir su conocimiento.

Estas prácticas me han ayudado a introducirme en el mundo laboral de una manera muy enriquecedora, debido a que he podido ver los diferentes ámbitos de la ciencia de datos y todos los pasos que hay en el proceso de un proyecto. Sin embargo no he podido ver a fondo la parte del desarrollo de los planes de proyecto y como realizar la administración de los recursos que también son una parte muy importante del mundo del *data science*. Además, estoy muy agradecido a Carlos y Juan por la libertad a la hora de elegir la estrategia para solucionar los problema y poder investigar nuevas soluciones, así como, a Ana Gentil por su comprensión y preocupación por la integración con el equipo. Además de a todos mis compañeros por su ayuda durante toda la estancia en prácticas.



# Capítulo 3

## Memoria TFG

### 3.1. Introducción

Esta memoria comienza con unos conceptos previos destinados al álgebra multilineal que serán imprescindibles para el desarrollo del cálculo tensorial necesario para la manipulación de los tensores propios del tema de estudio: tensor métrico y tensor curvatura de Riemann.

Seguidamente pasaremos a la introducción de las variedades diferenciables, los fibrados tangentes y cotangentes e introduciremos los campos vectoriales y tensoriales sobre variedades diferenciables. Esto nos permitirá definir una variedad semi-riemanniana como aquella que está dotada de un campo tensorial, conocido como tensor métrico, de tipo  $(0,2)$  que cumple las propiedades de simétrico y no degenerado.

En esta variedad introduciremos el concepto de conexión de Levi-Civita, con la que estudiaremos finalmente las ecuaciones de las curvas geodésicas y la expresión de las componentes del tensor curvatura de Riemann, para finalizar con el tensor de Ricci.

### 3.2. Conceptos previos

En esta sección se van a mostrar algunos conceptos previos que van a ser necesarios para poder entender este trabajo. Primero explicaremos algunos conceptos básicos sobre álgebra multilineal y, posteriormente, otros sobre el cálculo tensorial.

### 3.2.1. Conceptos básicos de álgebra lineal y multilineal

**Definición 3.2.1** (Ver [9]). Sea  $V$  un espacio vectorial sobre un cuerpo  $K$ . El **espacio vectorial dual** de  $V$  es el conjunto de todas las aplicaciones lineales  $\varphi : V \rightarrow K$ . Se denotará como  $V^*$ . A los elementos de  $V^*$  se les llamará **covectores** o **uno-formas**. También se le puede denotar como  $\text{Hom}_K(V, K)$ .

**Definición 3.2.2** (Ver [9]). Sea  $V$  un  $K$ -espacio vectorial y sea  $V^*$  el dual. Sea  $\beta = \{e^i\}_{i=1}^n$  una base de  $V$ , entonces denotamos por  $\beta^* = \{e_i^*\}_{i=1}^n$  la base del espacio dual  $V^*$  que cumple que

$$e_j^*(e^i) = \delta_j^i, \quad \forall i, j \in \{1, \dots, n\}$$

siendo  $\delta_j^i$  la delta de Kronecker,

$$\delta_j^i := \begin{cases} 0 & \text{si } i \neq j \\ 1 & \text{si } i = j. \end{cases}$$

**Definición 3.2.3** (Ver [9]). Una **aplicación multilineal** o **forma multilineal**, es una aplicación de un espacio vectorial,  $V^n$  a otro espacio vectorial  $W$ , de dimensión  $m$ .

$$\begin{aligned} f : V^n &\rightarrow W^m \\ (v_1, \dots, v_n) &\rightarrow f((v_1, \dots, v_n)) = (f_1((v_1, \dots, v_n)), \dots, f_m((v_1, \dots, v_n))) \end{aligned} \quad (3.1)$$

con  $f_k(v_1, \dots, v_i, \dots, v_n)$  aplicaciones lineales con respecto a la variable  $v_i$  para cualquier  $i \in \{1, \dots, n\}$  y para cualquier  $k \in \{1, \dots, m\}$ . En el caso particular en el que  $m = 1$  la aplicación  $f$  será llamada **aplicación lineal**.

## Tensores

En esta subsección se van a mostrar conceptos básicos sobre los tensores y el cálculo tensorial.

**Definición 3.2.4** (Ver [3]). Sea  $V$  un espacio vectorial real. Un **tensor  $k$  veces covariante** es cualquier aplicación multilineal  $T : \times_{i=1}^k V \rightarrow \mathbb{R}$ . Al conjunto de todos los  $k$ -tensores se denota como  $\mathcal{T}^k(V)$

**Ejemplo 3.2.5.** Para visualizar esto vamos a poner un ejemplo de tensor 3-covariante en  $\mathbb{R}^3$  dado por el determinante de una matriz  $3 \times 3$ .

$$T : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \mapsto \mathbb{R}, \quad (\vec{x}, \vec{y}, \vec{z}) \mapsto T((\vec{x}, \vec{y}, \vec{z})) := \det(\vec{x}, \vec{y}, \vec{z})$$

Es decir que:

$$T \left( \left( \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}, \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} \right) \right) = \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}$$

Usando las propiedades del determinante es fácil darse cuenta que  $T \in \mathcal{T}^3(\mathbb{R})$ , esto es, que  $T$  es una aplicación multilineal.

**Definición 3.2.6** (Ver [3]). Sea  $V$  un espacio vectorial real de dimensión  $n$ , y sea  $V^*$  su espacio dual. Definiremos un **tensor  $r$  veces contravariante** a una aplicación multilineal  $T : \times_{i=1}^r V^* \rightarrow \mathbb{R}$

**Definición 3.2.7** (Ver [3]). Sea  $V$  un espacio vectorial real y sea  $V^*$  su espacio vectorial dual. Entonces, un **tensor  $r$  veces contravariante y  $s$  veces covariante** será una aplicación multilineal de la forma:

$$T : \times_{i=1}^r V^* \times \times_{j=1}^s V \rightarrow \mathbb{R}$$

Se denotará como un tensor de tipo  $(r,s)$ .

**Proposición 3.2.8** (Ver [3]). Toda aplicación multilineal del tipo  $T : \times_{i=1}^n V \rightarrow \mathbb{R}^m$  se puede transformar a un tensor de tipo  $T : \times_{i=1}^n V \rightarrow \mathbb{R}$  mediante el uso de elementos del espacio dual.

**Definición 3.2.9** (Ver [8]). Dados dos tensores  $T$  y  $S$  de rangos  $k$  y  $m$ , entonces se define el **tensor producto** como el tensor  $(k+m)$ -tensor dado por:

$$T \otimes S(v_1, \dots, v_k, v_{k+1}, \dots, v_{k+m}) := T(v_1, \dots, v_k) \times S(v_{k+1}, \dots, v_{k+m})$$

## Convenio de Einstein

Esta notación se usa para simplificar la notación del álgebra tensorial. En resumen, consiste en que cuando se repite un índice arriba y abajo entonces suponemos que sumamos sobre él. Por ejemplo la combinación lineal de una serie de vectores se puede representar de esta forma:

$$\lambda^1 \cdot \vec{x}_1 + \dots + \lambda^n \cdot \vec{x}_n = \lambda^i \cdot \vec{x}_i \quad (3.2)$$

O un tensor  $(1,3)$  tal que  $R : V^* \times V \times V \times V \rightarrow \mathbb{R}$  con el vector del dual  $a = (a_1, a_2, a_3, a_4)$  y los vectores  $b = (b^1, b^2, b^3, b^4)$ ,  $c = (c^1, c^2, c^3, c^4)$  y  $d = (d^1, d^2, d^3, d^4)$

$$\sum_{i=1}^4 \sum_{j=1}^4 \sum_{k=1}^4 \sum_{l=1}^4 R_{ijkl}^i a_i \cdot b^j \cdot c^k \cdot d^l = R_{ijkl}^i a_i \cdot b^j \cdot c^k \cdot d^l \quad (3.3)$$

En este trabajo se utilizará siempre el convenio de Einstein

### 3.2.2. Conceptos básicos de topología

**Definición 3.2.10.** Sea  $(X, \mathcal{T})$  un espacio topológico. Diremos que  $(X, \mathcal{T})$  es un espacio topológico es Hausdorff o  $T_2$  si  $\forall p, q \in X$  con  $p \neq q \exists \mathcal{U}_p \in \mathcal{N}_p$  y  $\mathcal{V}_q \in \mathcal{N}_q$  tales que  $\mathcal{U}_p \cap \mathcal{V}_q = \emptyset$ .

**Definición 3.2.11.** Diremos que un espacio topológico  $(X, \mathcal{T})$  es **localmente euclídeo** si para todo  $p \in X$  existe un entorno  $\mathcal{U}$  de  $p$  tal que  $\exists \varphi : U' \subseteq \mathbb{R}^n \rightarrow \mathcal{U}$  que es un homeomorfismo.

**Definición 3.2.12.** Sean  $X$  e  $Y$  dos espacios topológicos. Diremos que una aplicación  $f : X \rightarrow Y$  es un **homeomorfismo** si  $f$  es una aplicación biyectiva y  $f^{-1}$  es una aplicación continua.

**Definición 3.2.13.** sea  $(X, \mathcal{T})$  un espacio topológico. Decimos que  $\mathcal{B} \subset \mathcal{P}(X)$  es una **base** de  $\mathcal{T}$  si:

1.  $\mathcal{B} \subset \mathcal{T}$

2. Para todo  $A \in \mathcal{T}$  podemos encontrar  $\{U_i : i \in I\} \subset \mathcal{B}$ , tal que  $\bigcup_{i \in I} U_i = A$ .

**Definición 3.2.14.** Sea  $(X, \mathcal{T})$  un espacio topológico y sea  $x \in X$ . Una **base de entornos** de  $x$  es un conjunto  $\mathcal{B}_x \subset \mathcal{N}_x \subset \mathcal{P}(X)$  tal que para todo entorno  $V$  de  $x$ , hay  $B \in \mathcal{B}_x$  con  $B \subset V$

**Definición 3.2.15.** Sea  $(X, \mathcal{T})$  un espacio topológico:

- Decimos que  $(X, \mathcal{T})$  satisface el **primer axioma de contabilidad**, o que es **1AN**, si para todo punto  $p \in X$  existe una base de entornos contable.
- Decimos que  $(X, \mathcal{T})$  satisface el **segundo axioma de contabilidad** si  $\mathcal{T}$  tiene una base contable.

### 3.3. Variedades diferenciables

El concepto de superficie plana ha sido una idea clásica. Ya en los elementos de Euclides se define como: "Una superficie plana es aquella que yace por igual respecto de las líneas que están en ella". Este libro sería la base de la geometría durante más de 2000 años, y no existiría otro tipo de geometría.

En el siglo XVIII con el nuevo espíritu reformista que se instauró en la sociedad y las semillas que había dejado sembradas René Descartes, un siglo antes, se comenzaron a estudiar y cuestionar las bases del sistema. Algunos de los matemáticos que comenzaron este profundo cambio fueron, por ejemplo, el italiano Giovanni Saccheri (1667 -1733) y el suizo Johann Lambert (1728 - 1777).

Posteriormente a inicios del siglo XIX otro gran matemático sentó las bases para el estudio de la geometría de las superficies dos-dimensionales: Karl Friedrich Gauss, con el artículo de 1827 "Investigación de superficies curvadas". Posteriormente en la década de 1830 otros grandes matemáticos, como pueden ser: Janos Bolyai (1802 - 1860) o Nikolai Lobachevsky (1792 - 1856) crearon un nuevo tipo de geometría. De todo esto se hablará en secciones futuras.

Además, a todas estas nuevas ideas se añadía la necesidad de generalizarlas a un número mayor de dimensiones. Por ejemplo, el matemático alemán Carl Gustav Jakob Jacobi para calcular el volumen de un esfera n-dimensional. A mediados del siglo XIX se formalizó esa generalización de la geometría de más de dos dimensiones, con matemáticos de la talla de Herman Grassman en su libro *Lineale Ausdhungslehre* de 1844 que ya presenta unas bases conceptuales para la geometría n-dimensional, o Cauchy y Gauss<sup>1</sup> que lo generalizaron para  $\mathbb{R}^n$ .



Figura 3.1: Retratos de Bernhard Riemann y Karl Friedrich Gauss

---

<sup>1</sup>En sus notas de la conferencia, donde ya presenta la generalización de las superficies bajo la palabra "manigfaltigkeiten" cuya traducción al castellano es "variedad".

En 1854 un joven Bernhard Riemann en su *habilitationsvortrag* o en castellano *lección de habilitación* presentó la idea de "*n-dimensionale mannigfaltigkeiten*" o *variedad n-dimensional*, un objeto matemático que no pertenecía a ningún marco de las matemáticas, estaba a caballo entre la matemática, la física y la filosofía. En ellas dio diferentes enfoques a este nuevo objeto, basándose en la teoría de conjuntos, variedades discretas, o en topología, variedades continuas.

Sin embargo la idea de variedad que mostró Riemann en su lección de habilitación no es la que hoy en día usamos. Tras la gran crisis de los fundamentos de las matemáticas de finales del siglo XX y su necesidad de dotar de un formalismo a este campo se comenzaron a redefinir los objetos matemáticos respecto de unos axiomas. Ya en el siglo XX Veblen y su alumno J.H.C. Whitehead dotaron de una axiomatización al concepto de variedad; es la definición que vemos hoy en día.

**Definición 3.3.1** (Ver [12]). *Una variedad ó variedad topológica  $\mathcal{M}$  de dimensión  $n$ , ó una  $n$ -variedad, es un espacio topológico con las siguientes propiedades:*

1.  $\mathcal{M}$  es Hausdorff.
2.  $\mathcal{M}$  es localmente euclídeo de dimensión  $n$ .
3.  $\mathcal{M}$  satisface el segundo axioma de numerabilidad, es  $2AN$ .

Riemann, en su clases de habilitación, presentó la idea de dotar de una estructura a las variedades para poder operar con ellas. En nuestro caso nos centraremos en las estructuras diferenciables, es decir, un conjunto de aplicaciones continuas, inyectivas y diferenciables de  $\mathbb{R}^n$  con la que equiparemos a nuestra variedad,  $\mathcal{M}$ , creando un nuevo tipo de variedades, variedades diferenciables.

**Definición 3.3.2** (Ver [5]). *Una variedad topológica,  $\mathcal{M}$ , es una **variedad diferenciable** si está dotada de una estructura diferenciable. Es decir, existe un conjunto de aplicaciones inyectivas  $X = \{ x_\alpha : U_\alpha \subset \mathbb{R}^n \rightarrow M \text{ con } U_\alpha \subset \mathbb{R}^n \text{ abierto} \}_{\alpha \in I}$  cumpliéndose que:*

1.  $\bigcup_{\alpha \in I} x_\alpha(U_\alpha) = M$
2. Para todo par  $\alpha, \beta \in I$  con  $x_\alpha(U_\alpha) \cap x_\beta(U_\beta) = W \neq \emptyset$  la aplicación  $x_\beta^{-1} \circ x_\alpha$  es diferenciable. (Fig:3.3)
3. La familia  $\{(U_\alpha, x_\alpha)\}_{\alpha \in I}$  es maximal relativa a las condiciones 1. y 2.

Lo que enuncia la tercera proposición es que si tenemos una estructura diferencial

$$\mathcal{V} = \{(U_i, x_i)\}_{i \in I}$$



de  $\mathcal{M}$  y una parametrización  $(U_0, \mathbf{x}_0)$  de  $\mathcal{M}$  tal que  $\mathbf{x}_0^{-1} \circ \mathbf{x}_i$  y  $\mathbf{x}_i^{-1} \circ \mathbf{x}_0$  es de clase  $\mathcal{C}^\infty$  para todo  $\mathbf{x}_i$  de  $\mathcal{V}$  entonces ocurre que  $(U_0, \mathbf{x}_0)$  pertenece a  $\mathcal{V}$ .

A cualquier familia  $\{U_i, \mathbf{x}_i\}_{i \in I}$  que satisface las dos primera proposiciones de la definición anterior se le llama **atlas** de  $\mathcal{M}$  y si también satisface la tercera proposición se le llamará **atlas maximal**.

Puede ocurrir que una variedad no tenga ninguna estructura diferenciable compatible o que tenga varias. En este último caso diremos que dos atlas son **equivalentes** si ambos definen la misma estructura diferenciable.

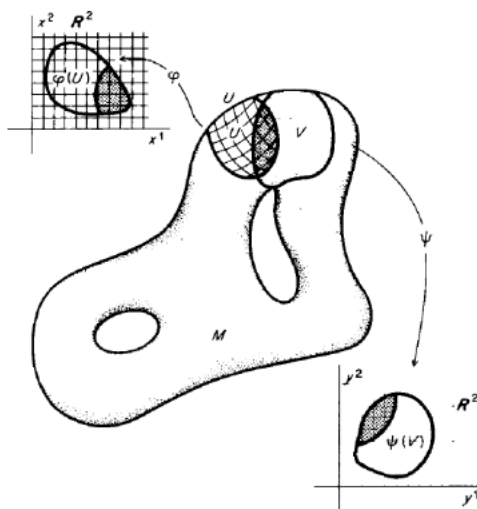


Figura 3.2: Ejemplo de cartas de una variedad y su intersección.[12]

Para poder ilustrar el ejemplo de una variedad vamos a mostrar que una esfera n-dimensional es una variedad n-dimensional.

**Ejemplo 3.3.3** (Ver [13]). Sea  $\mathcal{S}^n \subset \mathbb{R}^{n+1}$  tal que

$$\mathcal{S}^n = \left\{ \vec{x} = (x_1, \dots, x_{n+1}) \in \mathbb{R}^{n+1} : \sum_{i=1}^{n+1} x_i^2 = 1 \right\}.$$

Se definen los abiertos coordenados de  $\mathbb{R}^n$  como:

$$U_i^+ = \{ \vec{x} = (x_1, \dots, x_i, \dots, x_{n+1}) \in \mathcal{S}^n : x_i > 0 \}$$

$$U_i^- = \{ \vec{x} = (x_1, \dots, x_i, \dots, x_{n+1}) \in \mathcal{S}^n : x_i < 0 \}$$

Por consiguiente que para todo punto de  $\mathcal{S}^n$  pertenecerá a las cartas  $U_i^+$  ó  $U_i^-$ . De manera que

se tendrá que  $2I + 2$  cartas que satisficieran el primer requisito.

$$\mathcal{S}^n \subset \bigcup_{i=0}^I U_i^+ \cup \bigcup_{i=0}^I U_i^-$$

Después, se buscan los homeomorfismos coordenados. Que cumplan la segunda propiedad de la definición 3.3.2.  $\mathbf{x}_i^- = \mathbf{x}_i^+ : U_i \subseteq \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$

$$(x_1, \dots, x_{n+1}) \rightarrow \mathbf{x}_i(x_1, \dots, x_{n+1}) = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_{n+1})$$

Como se observa es la proyección  $i$ -ésima de  $\mathbb{R}^{n+1}$  a  $\mathbb{R}^n$ , que como bien se vio en la asignatura de geometría diferencial y topología son continuas e inyectivas.

Los homeomorfismos inversos son:

$$\begin{aligned} (\mathbf{x}_i^+)^{-1}(x_1, \dots, x_n) &= (x_1, \dots, x_{i-1}, \sqrt{1 - (x_1^2 + \dots + x_n^2)}, x_{i+1}, \dots, x_n) \\ (\mathbf{x}_i^-)^{-1}(x_1, \dots, x_n) &= (x_1, \dots, x_{i-1}, -\sqrt{1 - (x_1^2 + \dots + x_n^2)}, x_{i+1}, \dots, x_n) \end{aligned} \quad (3.4)$$

Como se puede observar estas funciones son continuas.

Similar a como ocurre en  $\mathbb{R}^3$  en las variedades también existen espacios tangentes a un punto. Si recordamos el concepto de plano tangente a un punto  $p$  de una superficie es el conjunto de vectores que son tangentes a la superficie en el punto, pero en el caso de las variedades el concepto de vector tangente no está definido de forma tan natural, por ello es necesario introducirlo.

**Definición 3.3.4** (Ver [8]). Sean  $\mathcal{M}$  y  $\mathcal{N}$  dos variedades diferenciables de dimensiones  $n$  y  $m$ , respectivamente. Una aplicación  $f : \mathcal{M} \rightarrow \mathcal{N}$  se dice que es **diferenciable** o de clase  $\mathcal{C}^\infty$  en un punto  $p \in \mathcal{M}$  si existe una parametrización  $(U, \psi)$  de  $\mathcal{M}$  en el punto  $p$  y otra parametrización  $(V, \phi)$  de  $\mathcal{N}$ , con  $f(\psi(U)) \subset \phi(V)$ :

$$f := \psi^{-1} \circ f \circ \phi : U \subset \mathbb{R}^m \rightarrow \mathbb{R}^n$$

es diferenciable. Diremos que la aplicación  $f$  es diferenciable en un subconjunto de  $\mathcal{M}$  si lo es en todo punto del conjunto. Al conjunto de todas las funciones diferenciables entre las variedades diferenciables  $\mathcal{M}$  y  $\mathcal{N}$  como  $\mathcal{C}^\infty(\mathcal{M}, \mathcal{N})$

**Definición 3.3.5** (Ver [5]). Sea  $\mathcal{M}$  una variedad  $n$ -dimensional, sea  $\alpha : ]-\varepsilon, \varepsilon[ \rightarrow \mathcal{M}$  una aplicación diferenciable y  $\alpha(0) = p \in \mathcal{M}$ . Sea  $\mathcal{D}$  el conjunto de funciones de  $\mathcal{M}$  diferenciables en  $p$ . El vector tangente a la curva  $\alpha$  en  $t = 0$  es una función  $\dot{\alpha}(0) : \mathcal{D} \rightarrow \mathbb{R}$  dada por

$$\dot{\alpha}(0)f = \left. \frac{d(f \circ \alpha)}{dt} \right|_{t=0}, \quad f \in \mathcal{D}.$$

Un **vector tangente** en  $p$  es el vector tangente en  $t = 0$  para alguna curva  $\alpha : ]-\varepsilon, \varepsilon[ \rightarrow \mathcal{M}$  con  $\alpha(0) = p$ . El conjunto de todos los vectores tangentes en  $p$  será denotado en lo que sigue por  $T_p\mathcal{M}$ .

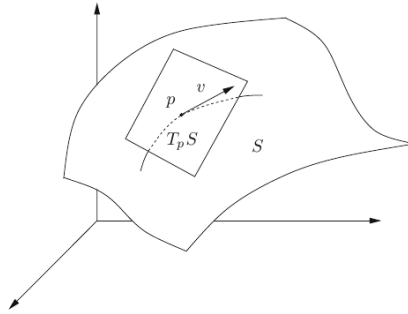


Figura 3.3: Espacio tangente de una superficie.

Esta idea tan abstracta de vector tangente en un punto no es muy operativa. Para poder manejar este concepto vamos a usar los atlas de la variedad que transforma nuestra variedad en un objeto de  $\mathbb{R}^n$  que si que conocemos bien. Para poder operar con ello vamos a necesitar una parametrización de  $\mathcal{V} \subseteq \mathcal{M}$ , con  $p \in \mathcal{V}$ ,  $x : U \subseteq \mathbb{R}^n \rightarrow \mathcal{V} \subseteq \mathcal{M}$  tal que  $x(0) = p$ . Dada esta carta  $x : U \rightarrow \mathcal{M}$  y  $f \in \mathcal{D}$ , podemos denotar por

$$\tilde{f} := f \circ x, \quad x^{-1} \circ \alpha(t) = (x^1(t), \dots, x^n(t))$$

Entonces, restringiendo  $f$  a  $\alpha$ , obtenemos que

$$\dot{\alpha}(0)f = \left. \frac{d(f \circ \alpha)}{dt} \right|_{t=0} = \left. \frac{d\tilde{f}(x^1(t), \dots, x^n(t))}{dt} \right|_{t=0} = \dot{x}^i(0) \frac{\partial \tilde{f}(0)}{\partial x^i} = \dot{x}^i(0) \left( \frac{\partial}{\partial x^i} \right)_0 \tilde{f} \quad (3.5)$$

donde hemos usado el convenio de suma de índices repetidos de Einstein. Dicho de otra forma, el vector  $\dot{\alpha}(0)$  puede ser expresado en la parametrización dada por la carta  $x : U \rightarrow \mathcal{M}$  por

$$\dot{\alpha}(0) = \dot{x}^i(0) \left( \frac{\partial}{\partial x^i} \right)_0$$

donde  $\left( \frac{\partial}{\partial x^i} \right)_0$  es el vector tangente en  $p$  a la  $i$ -ésima “curva coordenada”,

$$\alpha : ]-\varepsilon, \varepsilon[ \rightarrow \mathcal{M}, \quad x_i \mapsto x(0, \dots, 0, x_i, 0, \dots, 0).$$

La expresión (3.5) muestra que el vector tangente a la curva  $\alpha$  en  $p$  depende solamente de la derivada de  $\alpha$  en el sistema de coordenadas dado por la carta  $x : U \rightarrow \mathcal{M}$ . Se deduce también de (3.5) que el conjunto  $T_p\mathcal{M}$ , con las operaciones usuales en funciones, tiene estructura de espacio vectorial real de dimensión  $n$ , y que la parametrización asociada a la carta  $x : U \rightarrow \mathcal{M}$  determina una base asociada  $\left\{ \left( \frac{\partial}{\partial x^1} \right)_0, \dots, \left( \frac{\partial}{\partial x^n} \right)_0 \right\}$  en  $T_p\mathcal{M}$ . El espacio vectorial  $T_p\mathcal{M}$  será llamado como **espacio tangente** de  $\mathcal{M}$  en  $p$ . Todo esto puede ser resumido como sigue en la siguiente proposición.

**Proposición 3.3.6** (Ver [14]). *Sea  $\mathcal{M}$  una variedad diferenciable de dimensión  $n$ . Sea  $p \in \mathcal{M}$  un punto de  $\mathcal{M}$ . El espacio tangente  $T_p\mathcal{M}$  en el punto  $p$  es un espacio vectorial de dimensión  $n$ . Dada la carta  $x : U \rightarrow \mathcal{M}$ , podemos elegir como base de  $T_p\mathcal{M}$*

$$\left\{ \left( \frac{\partial}{\partial x^1} \right)_p, \dots, \left( \frac{\partial}{\partial x^n} \right)_p \right\},$$

siendo  $\left( \frac{\partial}{\partial x^i} \right)_p$  el vector tangente a la curva

$$x_i \mapsto x(x^{-1}(p) + (0, \dots, 0, x_i, 0, \dots, 0))$$

en el punto  $p$ .

**Proposición 3.3.7** (Ver [14]). *Sea  $\mathcal{M}$  una variedad diferenciable y sean  $\{U, x\}$  y  $\{V, y\}$  dos cartas de  $\mathcal{M}$ , sea  $p \in x(U) \cap y(V)$ , y sea  $v \in T_p\mathcal{M}$ . Entonces, como hemos visto,  $v$  respecto de estas cartas tendrá las siguientes expresiones*

$$v_x = v_x^i \frac{\partial}{\partial x^i} \Big|_p, \quad v_y = v_y^j \frac{\partial}{\partial y^j} \Big|_p.$$

Llegados a estas dos expresiones la pregunta clave, sería ¿Cómo están relacionados  $\{v_x^i\}$  y  $\{v_y^j\}$ ? Como esto puede ser consultado en [14], sea  $f \in C^\infty(p)$ , sea  $\alpha$  una curva con  $\alpha(0) = p$  y  $\dot{\alpha}(0) = \vec{v}$ . Entonces,

$$\begin{aligned} v(f)(p) &= \dot{\alpha}(0)(f) = \frac{d}{dt} f(\alpha(t)) \Big|_{t=0} = \frac{d}{dt} f(x(x^{-1}(\alpha(t)))) \Big|_{t=0} \\ &= \frac{d}{dt} (f \circ x) \circ (x^{-1} \circ \alpha)(t) \Big|_{t=0} = \frac{d}{dt} (f \circ x) \circ (x^1(t), \dots, x^n(t))(t) \Big|_{t=0} \\ &= \frac{\partial}{\partial x^i} (f \circ x)(p) \cdot \dot{x}^i(0) = \dot{x}^i(0) \frac{\partial}{\partial x^i} \Big|_p f \circ x. \end{aligned}$$

Lo que implica que  $v_x^i$  es la coordenada  $i$ -ésima del vector  $\frac{d}{dt} x^{-1}(\alpha(t)) \Big|_{t=0}$ . Similarmente,  $v_y^j$  es la coordenada  $j$ -ésima del vector  $\frac{d}{dt} y^{-1}(\alpha(t)) \Big|_{t=0}$ . Como  $(x^1(t), \dots, x^n(t)) = x^{-1}(\alpha(t))$  y

$(y^1(t), \dots, y^n(t)) = y^{-1}(\alpha(t))$  tenemos que

$$\begin{aligned} x^i(t) &= (x^{-1}(\alpha(t)))^i = (x^{-1}(y(y^{-1}(\alpha(t))))^i \\ &= (x^{-1} \circ y(y^{-1}(\alpha(t))))^i = (x^{-1} \circ y(y^1(t), \dots, y^n(t)))^i \end{aligned}$$

Así queda que

$$v_x^i = \dot{x}^i(0) = \left. \frac{\partial (x^{-1} \circ y)^i}{\partial y^j} \right|_{(y^1(0), \dots, y^n(0))} \quad \dot{y}^j(0) = \left. \frac{\partial (x^{-1} \circ y)^i}{\partial y^j} \right|_{(y^1(0), \dots, y^n(0))} v_y^j$$

Haciendo un abuso de notación denotaremos  $\left(\frac{\partial}{\partial x^i}\right)_p$  por  $\frac{\partial}{\partial x^i}$  y  $\left(\frac{\partial}{\partial y^i}\right)_p$  por  $\frac{\partial}{\partial y^i}$ , la matriz  $\left.\frac{\partial (x^{-1} \circ y)^i}{\partial y^j}\right|_{(y^1(0), \dots, y^n(0))}$  será denotada por  $\frac{\partial x^i}{\partial y^j}$  y que su inversa será denotada por  $\frac{\partial y^j}{\partial x^i}$ . Quedando

$$\frac{\partial x^i}{\partial y^k} \frac{\partial y^k}{\partial x^j} = \delta_j^i,$$

Todo esto nos conduce a la ecuación

$$\frac{\partial}{\partial x^i} = \frac{\partial y^j}{\partial x^i} \frac{\partial}{\partial y^j} \quad (3.6)$$

**Definición 3.3.8** (Ver [14]). Al espacio dual de  $T_p\mathcal{M}$  se le llamará espacio cotangente, y se denotará como  $T_p\mathcal{M}^*$ . Y será:

$$T_p\mathcal{M}^* := \{\varphi : T_p\mathcal{M} \rightarrow \mathbb{R} : \varphi \text{ es una aplicación lineal}\}$$

Como dada una carta  $x : U \rightarrow \mathcal{M}$  tenemos la base asociada a  $T_p\mathcal{M}$  dada por  $\left\{\frac{\partial}{\partial x^1}, \dots, \frac{\partial}{\partial x^n}\right\}$ , tenemos asociada también la base dual que denotaremos por

$$\{dx^1, \dots, dx^n\}$$

Cumpliendo que

$$dx^i \left( \frac{\partial}{\partial x^j} \right) = \delta_j^i.$$

Si tenemos ahora la parametrización dada por la carta  $y : V \rightarrow \mathcal{M}$ , con  $p \in x(U) \cap y(V) \neq \emptyset$ , tendremos la correspondiente base de  $T_p\mathcal{M}^*$  dada por  $\{dy^1, \dots, dy^n\}$ . Como sabemos por (3.6) como se relacionan las correspondientes bases de  $T_p\mathcal{M}$  podemos escribir que

$$dy^k \left( \frac{\partial}{\partial x^i} \right) = \frac{\partial y^k}{\partial x^i} \delta_j^k = \frac{\partial y^k}{\partial x^i}$$

Por lo que nos queda el siguiente cambio de base para el espacio cotangente  $T_p\mathcal{M}^*$ ,

$$dy^k = \frac{\partial y^k}{\partial x^i} dx^i \quad (3.7)$$

**Definición 3.3.9** (Ver [8]). A la unión de todos los espacio es tangente de  $\mathcal{M}$  se denota como  $T(\mathcal{M})$ :

$$T(\mathcal{M}) = \bigcup_{p \in \mathcal{M}} T_p \mathcal{M} \quad (3.8)$$

A este espacio se le llama **fibrado tangente**. Y es posible dotarlo de una estructura diferenciable de dimensión  $2n$ .

Para poder visualizar esta idea de vector tangente a una variedad vamos a usar el ejemplo de la esfera de dimensión 2.

**Ejemplo 3.3.10.** Continuando con el ejemplo 3.3.3 pero tomando  $n = 2$  obtenemos que  $S^2 \subset \mathbb{R}^3$  vamos a estudiar el plano tangente a la esfera. Los conjuntos de son:

$$\begin{aligned} U_x^+ &= \{\vec{v} = (x, y, z) \in \mathcal{S}^2 \cdot : x > 0\} & U_x^- &= \{\vec{v} = (x, y, z) \in \mathcal{S}^2 \cdot : x < 0\} \\ U_y^+ &= \{\vec{v} = (x, y, z) \in \mathcal{S}^2 \cdot : y > 0\} & U_y^- &= \{\vec{v} = (x, y, z) \in \mathcal{S}^2 \cdot : y < 0\} \\ U_z^+ &= \{\vec{v} = (x, y, z) \in \mathcal{S}^2 \cdot : z > 0\} & U_z^- &= \{\vec{v} = (x, y, z) \in \mathcal{S}^2 \cdot : z < 0\} \end{aligned} \quad (3.9)$$

Y los homeomorfismos inversos son:

$$\begin{aligned} (\varphi_x^+)(y, z) &= (\sqrt{1 - y^2 - z^2}, y, z) \\ (\varphi_x^-)(y, z) &= (-\sqrt{1 - y^2 - z^2}, y, z) \\ (\varphi_y^+)(x, z) &= (x, \sqrt{1 - x^2 - z^2}, z) \\ (\varphi_y^-)(x, z) &= (x, -\sqrt{1 - x^2 - z^2}, z) \\ (\varphi_z^+)(x, y) &= (x, y, \sqrt{1 - y^2 - x^2}) \\ (\varphi_z^-)(x, y) &= (x, y, -\sqrt{1 - y^2 - x^2}) \end{aligned} \quad (3.10)$$

Entonces el espacio tangente a  $p \in \mathcal{S}^2$  será:

$$\begin{aligned} \frac{\partial(\varphi_x^+)}{\partial y} &= \left( \frac{-y}{\sqrt{1-y^2-z^2}}, 1, 0 \right) & \frac{\partial(\varphi_x^-)}{\partial y} &= \left( \frac{y}{\sqrt{1-y^2-z^2}}, 1, 0 \right) \\ \frac{\partial(\varphi_x^+)}{\partial z} &= \left( \frac{-z}{\sqrt{1-y^2-z^2}}, 0, 1 \right) & \frac{\partial(\varphi_x^-)}{\partial z} &= \left( \frac{z}{\sqrt{1-y^2-z^2}}, 0, 1 \right) \\ \frac{\partial(\varphi_y^+)}{\partial x} &= \left( 1, \frac{-x}{\sqrt{1-x^2-z^2}}, 0 \right) & \frac{\partial(\varphi_y^-)}{\partial x} &= \left( 1, \frac{x}{\sqrt{1-x^2-z^2}}, 0 \right) \\ \frac{\partial(\varphi_y^+)}{\partial z} &= \left( 0, \frac{-z}{\sqrt{1-x^2-z^2}}, 1 \right) & \frac{\partial(\varphi_y^-)}{\partial z} &= \left( 0, \frac{z}{\sqrt{1-x^2-z^2}}, 1 \right) \\ \frac{\partial(\varphi_z^+)}{\partial x} &= \left( 1, 0, \frac{-x}{\sqrt{1-x^2-y^2}} \right) & \frac{\partial(\varphi_z^-)}{\partial x} &= \left( 1, 0, \frac{x}{\sqrt{1-x^2-y^2}} \right) \\ \frac{\partial(\varphi_z^+)}{\partial y} &= \left( 0, 1, \frac{-y}{\sqrt{1-x^2-y^2}} \right) & \frac{\partial(\varphi_z^-)}{\partial y} &= \left( 0, 1, \frac{y}{\sqrt{1-x^2-y^2}} \right) \end{aligned} \quad (3.11)$$

Entonces cualquier punto  $p \in \mathcal{S}^2$  su espacio tangente podrá se expresado en una base. Por ejemplo supongamos que  $p = (0, 0, 1) \in U_z^+$  y por lo tanto  $T_{(0,0,1)}\mathcal{S}^2$  estará generado por:  $\{(0, 1, \frac{-y}{\sqrt{1-x^2-z^2}}), (1, 0, \frac{-x}{\sqrt{1-x^2-z^2}})\}$

### 3.4. El espacio euclídeo $\mathbb{R}^n$ como variedad diferenciable

Con el fin de estudiar ciertos objetos matemáticos, como puede ser el concepto de métrica o de conexión afín, vamos a centrarnos en  $\mathbb{R}^n$  como la variedad ambiente. En esta sección demostraremos que  $\mathbb{R}^n$  es una variedad diferenciable e introduciremos el concepto de campo vectorial sobre una variedad. Además, para poder operar con los campos vectoriales definiremos un operador, los corchete de Lie, y finalizaremos con el concepto de uno-forma y su diferencial.

**Proposición 3.4.1.** *El espacio  $\mathbb{R}^n$  puede ser dotado de estructura diferenciable convirtiéndose en una variedad diferenciable  $n$ -dimensional.*

*Demostración.* Se tiene que  $\mathbb{R}^n$  es un espacio métrico y por lo tanto Hausdorff y además cumple el segundo axioma de numerabilidad, y es localmente euclídeo como podemos ver con la aplicación identidad. Por lo tanto es una variedad topológica.

Una vez demostrado que  $\mathbb{R}^n$  es una variedad se va a demostrar que es una variedad diferenciable. Para ello se va a buscar un estructura diferenciable que sea compatible con  $\mathbb{R}^n$ . Dicha estructura será  $(U = \mathbb{R}^n, \mathbf{x} = Id)$ , con  $Id$  la aplicación identidad de  $\mathbb{R}^n$ . Para esto vamos a ver que dicha estructura es compatible comprobando que cumple las propiedades de la definición 3.3.2.

1.  $\mathcal{V} = \mathbb{R}^n$
2. Como la aplicación  $Id$  cumple claramente el segundo requisito ya que es un difeomorfismo.
3. La familia  $(\mathbb{R}^n, Id)$  es maximal, ya que siguiendo el razonamiento hecho en la demostración del apéndice B vemos que todo recubrimiento de  $\mathbb{R}^n$  va a estar incluido en la parametrización propuesta.

Por consecuente, podemos ver que  $\mathbb{R}^n$  es una variedad  $n$ -dimensional diferenciable. □

Una vez que ya hemos visto que  $\mathbb{R}^n$  es una variedad  $n$ -dimensional vamos a introducir diferentes conceptos para poder operar sobre las variedades.

**Definición 3.4.2** (Ver [12]). *Sea  $\mathcal{M}$  una variedad diferenciable. Un **campo vectorial**  $X$  es una correspondencia que asocia a cada punto  $p \in \mathcal{M}$  un vector de  $T_p\mathcal{M}$ .*

$$X : \mathcal{M} \rightarrow T\mathcal{M}, \quad \text{tal que } \pi \circ X = Id$$

Siendo  $\pi : T\mathcal{M} \rightarrow \mathcal{M}$  la proyección canónica. El campo será diferenciable si la aplicación  $X$  es diferenciable.

Pero como ocurría en el caso del vector tangente en un punto, para poder operar con estos conceptos necesitamos trabajar en un espacio conocido para nosotros y para ello utilizaremos las cartas de la variedad diferencial para expresar el campo vectorial.

Si tomamos una parametrización de  $\mathcal{M}$ ,  $x : U \subset \mathbb{R}^n \rightarrow \mathcal{M}$  podemos escribir  $X(p)$  respecto de dicha parametrización:

$$X(p) = a_i(p) \frac{\partial}{\partial x^i} \Big|_p$$

Con  $a_i : U \rightarrow \mathbb{R}$ , podemos observar que el campo  $X$  será diferenciable si las funciones  $a_i$  lo son, y al conjunto de todos los campos de clase  $\mathcal{C}^\infty$  sobre  $M$  lo denotaremos como  $\mathfrak{X}(\mathcal{M})$ .

Sea  $X \in \mathfrak{X}(\mathcal{M})$  y  $f \in \mathcal{C}^\infty(\mathcal{M})$  entonces  $Xf$ , es una función de valor real, tal que  $(Xf)(p) = X_p(f)$  para todo  $p \in \mathcal{M}$ . De la siguiente manera:

$$(Xf)(p) = a_i(p) \cdot \frac{\partial f}{\partial x^i} \Big|_p \quad (3.12)$$

A esta función  $Xf$  se le llamará derivada direccional de  $f$  sobre el campo vectorial  $X$ . Diremos que el campo es diferenciable si lo es para toda  $f \in \mathcal{C}^\infty(\mathcal{M})$ .

**Definición 3.4.3** (Ver [5]). *Sea  $\mathcal{M}$  una variedad diferenciables, sean  $\forall X, Y \in \mathfrak{X}(\mathcal{M})$  y  $f \in \mathcal{C}^\infty(\mathcal{M})$ . El corchete de Lie  $[X, Y]$  es una aplicación bilineal que actúa sobre  $f$  en un punto  $p \in \mathcal{M}$  definida como:*

$$\begin{aligned} [\cdot, \cdot] : \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) &\rightarrow \mathfrak{X}(\mathcal{M}) \\ [X, Y](f) &= X(Yf) - Y(Xf) \end{aligned}$$

**Proposición 3.4.4.** *Sea  $\mathcal{M}$  una variedad diferenciable y sean  $X, Y \in \mathfrak{X}(\mathcal{M})$ . Entonces,  $[X, Y] \in \mathfrak{X}(\mathcal{M})$ .*

*Demostración.* Sean  $X$  e  $Y$  dos campos vectoriales y  $v : \mathcal{U} \rightarrow \mathcal{M}$  una carta de  $\mathcal{M}$ . Vamos a demostrar que existe dicho campo vectorial  $[X, Y]$ . Primero vamos a definir  $X$  e  $Y$  respecto de la parametrización dada por  $v$ .

$$\begin{aligned} X &= a^i \frac{\partial}{\partial v^i} \\ Y &= b^j \frac{\partial}{\partial v^j} \end{aligned} \quad (3.13)$$

Entonces para toda  $f \in \mathcal{C}^\infty(\mathcal{M})$ :

$$\begin{aligned} XYf &= X(b^j \frac{\partial f}{\partial v^j}) = a^i \cdot b^j \frac{\partial^2 f}{\partial v^i \partial v^j} + a^i \frac{\partial b^j}{\partial v^i} \frac{\partial f}{\partial v^j} \\ YXf &= Y(a^i \frac{\partial f}{\partial v^i}) = a^i \cdot b^j \frac{\partial^2 f}{\partial v^j \partial v^i} + b^j \frac{\partial a^i}{\partial v^j} \frac{\partial f}{\partial v^i} \end{aligned} \quad (3.14)$$



Entonces sumando  $XYf - YXf$  obtenemos:

$$[X, Y] = a^i \frac{\partial b^j}{\partial v^i} \frac{\partial f}{\partial v^j} - b^j \frac{\partial a^i}{\partial v^j} \frac{\partial f}{\partial v^i} \quad (3.15)$$

Ya que  $\frac{\partial^2 f}{\partial v^j \partial v^i} = 0$  y cambiamos los índices en el segundo sumando y obtenemos que:

$$[X, Y]f = \left( a^i \frac{\partial b^j}{\partial v^i} - b^j \frac{\partial a^i}{\partial v^j} \right) \frac{\partial f}{\partial v^j} \quad (3.16)$$

Con esto hemos probado que existe un único campo vectorial definido por  $[X, Y]$ .  $\square$

**Proposición 3.4.5** (Ver [8]). *Para todo  $X, Y, Z \in \mathfrak{X}(\mathcal{M})$  la operación  $[\cdot, \cdot]$  cumplirá las siguientes propiedades con  $\forall a, b \in \mathbb{R}$  y  $\forall f, g \in C^\infty(\mathcal{M})$ :*

- **Linealidad**  $[aX + bY, Z] = a[X, Z] + b[Y, Z]$
- **Antisimétrica**  $[X, Y] = -[Y, X]$
- **Identidad de Jacobi**  $[X, [Y, Z]] + [Z, [X, Y]] + [Y, [Z, X]] = 0$
- **Regla Leibniz**  $[fX, gY] = f[gX, Y] + fXY(g) - gY(f)X$

Y el campo resultante mantendrá la diferenciabilidad si  $X$  e  $Y$  son diferenciables. [5]

**Definición 3.4.6.** [12] pg. 15 Sea  $\mathcal{M}$  una variedad diferenciable. Definimos una **uno-forma**,  $\theta$  en la variedad  $\mathcal{M}$  como una aplicación lineal que asigna a cada punto  $p \in \mathcal{M}$  un elemento del espacio  $T_p^*(\mathcal{M})$ ,  $\theta_p$ .

Sea  $\theta$  una uno-forma y sea  $X \in \mathfrak{X}(\mathcal{M})$  denotamos  $\theta X$  la función real en  $M$  que a cada punto de  $\mathcal{M}$  es el valor de  $\theta(p)$  en  $X_p$ . Se dirá que una uno-forma es diferenciable si lo es para todo  $X \in \mathfrak{X}(\mathcal{M})$ . Y el conjunto de todas la uno formas de  $\mathcal{M}$  se denotará como  $\mathfrak{X}^*(\mathcal{M})$ .

**Definición 3.4.7** (Ver [12]). La diferencial de  $f \in C^\infty$  es una uno forma,  $df$ , tal que:

$$\begin{aligned} df_p : T_p \mathcal{M} &\rightarrow \mathbb{R} \\ v &\rightarrow df_p(v) = v \cdot f \end{aligned} \quad (3.17)$$

Que es la derivada direccional de  $f$  en el punto  $p$  respecto de la dirección  $\vec{v}$ .

Si se tiene una variedad diferenciable  $\mathcal{M}$  y una carta  $(\mathcal{U}, x)$  entonces se puede expresar  $df_p \vec{v}$  respecto de dicha carta. Como  $\vec{v}$  se puede expresar respecto de dicha carta  $\vec{v} = v^i \frac{\partial}{\partial x^i} \Big|_p$  entonces  $df_p \vec{v}$  respecto de  $x$  será:

$$df_p \vec{v} = v^i \frac{\partial f \circ x^{-1}}{\partial x^i} (x(p))$$

Si tomamos las funciones coordenadas  $x^i : V \rightarrow \mathbb{R}$  entonces obtenemos las 1-formas coordenadas  $dx^i$  en  $V$  que satisface que:  $dx^i_p(\frac{\partial}{\partial x^j}) = \delta_j^i$  y por lo tanto las uno-formas formarán una base del espacio cotangente en  $p$ . Y cualquier elemento del campo tensorial en  $V$  se podrá escribir como  $\vec{v} = \vec{v}_i dx^i$

En particular  $df$  se podrá escribir de la siguiente forma:

$$(df)_p = \frac{\partial f \circ x^{-1}}{\partial x^i} (dx^i)_p$$

### 3.5. Tensor Métrico

Como hemos dicho en la introducción, durante más de 2000 años se tomó como única la geometría euclídea. Basada en los cinco postulados:

1. Dos puntos distintos cuales quiera determinan un segmento de recta.
2. Un segmento de recta se puede extender indefinidamente en una línea recta.
3. Se puede trazar una circunferencia dados un centro y un radio cualquiera.
4. Todos los ángulos rectos son iguales entre sí.
5. Si una línea recta corta a otras dos, de tal manera que la suma de los dos ángulos interiores del mismo lado sea menor que dos rectos, las otras dos rectas se cortan, al prolongarlas, por el lado en el que están los ángulos menores que dos rectos.

Este último postulado también se puede reescribir como: "Por un punto exterior a una recta, se puede trazar una única paralela". Este enunciado creó muchas discusiones, debido a que había matemáticos que decían que era un axioma y otros que decían que era una consecuencia de los 4 anteriores y por lo tanto se podía deducir. Esta disputa duró siglos, y trajo de cabeza a algunos de las matemáticos más importantes de época, como pueden ser Lambert y Saccheri, que se cuestionaron este 5º postulado.

Estos dos matemáticos con su cuestión sobre el quinto postulado dieron lugar a una nueva idea, la geometría no euclídea. Pero hasta la década de 1830 no se publicó ningún resultado sobre esta nueva geometría ,con los matemáticos Bolyai y Lobachevsky que decidieron eliminar este quinto postulado. Por lo tanto, había infinitas rectas que pasaban por un mismo punto. A esta geometría se le conoce como la **geometría hiperbólica**. Una cosa absurda para aquella época, y tardaron años en darse cuenta que ambas geometrías era compatibles y lo único que ocurría es que dependen de distintos sistemas axiomáticos.

Bernhard Riemann, en su lección de habilitación presentó el concepto de variedad desde distintos enfoques matemáticos pero con punto de vista analítico y no axiomático, como se habían hecho previamente la generalización de los conceptos geométricos a más dimensiones. Para dar este enfoque analítico presentó la idea, previamente usada por Gauss para el estudio de las superficies, de métrica en una variedad. Además, Riemann generalizó las ideas de Gauss de curvatura, para que estas propiedades únicamente dependían de la variedad y no del espacio en el que se encontraba.

En esta sección introduciremos el concepto de métrica semi-Riemanniana y Riemanniana. Además, se mostrarán algunos ejemplos de métricas semi-Riemanniana y Riemanniana, como puede ser la métrica hiperbólica, y algunos conceptos sobre tensores.

**Definición 3.5.1.** Sea  $\mathcal{M}$  una variedad diferenciable. Se define un **campo tensorial** de clase  $\mathcal{C}^\infty$  de tipo  $(r,s)$  en  $\mathcal{M}$  como una aplicación para cada punto  $p \in \mathcal{M}$  se le asigna un tensor de tipo  $(r,s)$  con los espacios  $T_p\mathcal{M}$  y  $T_p\mathcal{M}^*$  tales que cada carta de las componentes sean de clase  $\mathcal{C}^\infty$ .

**Definición 3.5.2** (Ver [5]). Sea  $\mathcal{M}$  un variedad diferenciable. Diremos que un tensor de tipo  $(0,2)$  es una **métrica semi-Riemanniana** definida sobre  $\mathcal{M}$  si cumple que:

1. **Simétrica**  $g(u,v) = g(v,u) \forall u, v \in T_p\mathcal{M}$
2. **No degenerada** si para todo  $v$  ocurre que  $g(u,v) = 0$  entonces  $u = 0$ .

Y diremos que una **métrica es Riemanniana** si además cumple que es definida positiva  $g(u,u) > 0 \forall u \neq 0 \in T_p\mathcal{M}$

**Definición 3.5.3** (Ver [12]). Sea  $\mathcal{M}$  una variedad diferenciable y sea el campo tensorial  $g$  una métrica semi-riemanniana sobre  $\mathcal{M}$ . Entonces, diremos que la variedad diferenciable  $\mathcal{M}$  es una variedad semi-riemanniana, y lo denotaremos como  $(\mathcal{M}, g)$ . Se definirá de forma análoga una variedad riemanniana pero siendo  $g$  una métrica riemanniana.

Sea  $(\mathcal{M}, g)$  una variedad semi-Riemanniana y sea  $x : U \subseteq \mathbb{R}^n \rightarrow \mathcal{M}$  una carta, tal que  $p = x(x_1, \dots, x_n) \in x(U)$  y  $\frac{\partial}{\partial x^i}(q) = dx_q(0, \dots, 1, 0, \dots, 0)$ . Entonces, el tensor métrico  $g$  puede ser expresado respecto a la parametrización asociada a la carta  $x : U \rightarrow \mathcal{M}$  como

$$g = g_{ij}(q) dx^i \otimes dx^j$$

Siendo  $g_{ij}(q)$  las componentes del tensor métrico dadas por

$$g_{ij}(q) = g\left(\frac{\partial}{\partial x^i}(q), \frac{\partial}{\partial x^j}(q)\right).$$

El tensor métrico nos permite clasificar los vectores tangentes en una variedad semi-riemanniana de la siguiente manera:

**Definición 3.5.4** (Ver [12]). Sea  $(M, g)$  una variedad semi-Riemanniana. Sea  $v \in T_p \mathcal{M}$  diremos:

- $v$  es espacial o spacelike si  $g(v, v) > 0$ .
- $v$  es nulo o lightlike si  $g(v, v) = 0$  y  $v \neq 0$ .
- $v$  es temporal o timelike si  $g(v, v) < 0$ .

Esta clasificación puede parecer un poco artificial pero tiene un sentido físico. Para poner en contexto vamos a ver la métrica de Minkowski, que posteriormente estudiaremos:

$$g = -c^2 dt \otimes dt + dx \otimes dx + dy \otimes dy + dz \otimes dz \quad (3.18)$$

Vemos que esta métrica tiene cuatro componentes, una espacial y tres temporales. Las componentes espaciales son iguales que la métrica euclídea al cuadrado mientras que la componente temporal tiene un coeficiente  $-c^2$  que marca claramente el signo del módulo. Si recordamos las clases de física del instituto había un principio de la relatividad que decía que ningún objeto puede viajar más rápido que la luz en el vacío. En esta métrica podemos ver que si la velocidad de un objeto es menor que la de la luz se producirá una geodésica temporal, si va a la velocidad de la luz será nula o si va a una velocidad mayor que la luz será espacial<sup>3</sup>. Esta ejemplificación se puede ver en la figura 3.4.

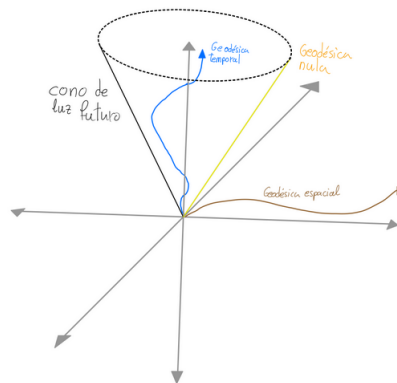


Figura 3.4: Ejemplo de un cono de luz futuro, donde se puede ver el cono de central que es el conjunto de geodésicas nulas, el interior del cono que es el conjunto de las geodésicas temporales y el exterior que es el conjunto de todas las geodésicas espaciales.

<sup>2</sup>Velocidad de la luz en el vacío

<sup>3</sup>El nombre de espacial viene a que únicamente el espacio puede expandirse más rápido que la velocidad de la luz en el vacío

### 3.5.1. Ejemplos de variedades riemannianas y semi-riemannianas

Algunos ejemplos de métrica son:

**Ejemplo 3.5.5. Métrica de Poincaré** o también llamada *métrica hiperbólica*. La variedad que tendremos será  $\mathcal{M} = \mathbb{R} \times \mathbb{R}^+$  definimos la métrica  $g = \frac{dx \otimes dx + dy \otimes dy}{y^2}$ . Entonces sus componentes serán en un punto  $p \in \mathcal{M}$ , con coordenadas  $p = (x, y)$ :

$$\begin{aligned} g_{11} &= g\left(\frac{\partial}{\partial x}(q), \frac{\partial}{\partial x}(q)\right) = \frac{1}{y^2} \\ g_{12} &= g\left(\frac{\partial}{\partial x}(q), \frac{\partial}{\partial y}(q)\right) = 0 \\ g_{21} &= g\left(\frac{\partial}{\partial y}(q), \frac{\partial}{\partial x}(q)\right) = 0 \\ g_{22} &= g\left(\frac{\partial}{\partial y}(q), \frac{\partial}{\partial y}(q)\right) = \frac{1}{y^2} \end{aligned} \tag{3.19}$$

Por lo tanto su matriz será:

$$G = (g_{ij}) = \begin{pmatrix} \frac{1}{y^2} & 0 \\ 0 & \frac{1}{y^2} \end{pmatrix} \tag{3.20}$$

Vamos a ver que la métrica de Poincaré es una métrica riemanniana observando la expresión matricial, en ella podemos observar que es definida positiva debido a que la variedad sobre la que está definida no tiene puntos que hagan que sea menor o igual que cero. Además será simétrica ya que los términos cruzados son iguales.

Por ejemplo, sea  $p = (1, 1)$  y sea  $\vec{v} = 3 \frac{\partial}{\partial x} \Big|_p + 5 \frac{\partial}{\partial y} \Big|_p$  entonces  $G(\vec{v}, \vec{v}) = 3^2 + 5^2 = 34$ .

O si tomamos  $p = (0, 5)$  y sea  $\vec{v} = 3 \frac{\partial}{\partial x} \Big|_p + 4 \frac{\partial}{\partial y} \Big|_p$  entonces  $G(\vec{v}, \vec{v}) = \frac{3^2 + 4^2}{5^2} = 1$ .

**Ejemplo 3.5.6. Métrica en una esfera dos dimensional** Sea la variedad diferenciable  $\mathcal{M} = S^2$  que está en  $\mathbb{R}^3$ . Tomamos la métrica euclídea con la carta identidad:

$$g = dx \otimes dx + dy \otimes dy + dz \otimes dz \tag{3.21}$$

Entonces tomamos el sistema de coordenadas esféricas tal que:

$$\begin{aligned} \mathcal{C} : \mathbb{R} \times [0, 2\pi] \times [0, \pi] &\rightarrow \mathbb{R}^3 \\ (r, \theta, \phi) \rightarrow C(r, \theta, \phi) &= (r \sin \theta \cos \phi, r \sin \theta \sin \phi, r \cos \theta) \end{aligned} \tag{3.22}$$

Vamos expresar la ecuación 3.21 respecto de la carta  $\mathcal{C}$ . Por lo tanto, como se vio en la proposición 3.6 obtenemos que:

$$\begin{aligned} dx &= (\text{sen}(\theta)\cos(\phi))dr + (r\cos(\theta)\sin(\phi))d\theta + (-r\text{sen}(\theta)\cos(\phi))d\phi \\ dy &= (\text{sen}(\theta)\text{sen}(\phi))dr + (r\cos(\theta)\sin(\phi))d\theta + (r\text{sen}(\theta)\cos(\phi))d\phi \\ dz &= \cos(\theta)dr + r\sin(\theta)d\theta \end{aligned} \quad (3.23)$$

Entonces sustituimos en la ecuación 3.21 y obtenemos la siguiente métrica respecto de la cara  $\mathcal{C}$  y obtenemos:

$$g = dr \otimes dr + r^2 d\theta \otimes d\theta + r^2 \sin^2(\theta) d\phi \otimes d\phi \quad (3.24)$$

Finalmente si tomamos el radio como una constante, por lo tanto  $dr = 0$ , se obtiene que la métrica euclídea en una dos esfera es de la forma:

$$g = r^2 d\theta \otimes d\theta + r^2 \sin^2(\theta) d\phi \otimes d\phi. \quad (3.25)$$

Sea  $x : (0, 2\pi) \times (0, \pi) \rightarrow \mathcal{S}^2$  una carta de la esfera  $\mathcal{S}^2$  que asigna a cada punto de  $\mathcal{S}^2$  su correspondiente ángulo  $\theta$  y su correspondiente ángulo  $\phi$ . La expresión del tensor métrico respecto a esta carta quedaría como

$$g = \begin{pmatrix} R^2 & 0 \\ 0 & R^2 \sin^2(\theta) \end{pmatrix}$$

Siendo  $R \in \mathbb{R}^+$  el radio considerado para la esfera  $\mathcal{S}^2$ . Esta métrica es un ejemplo de métrica Riemanniana, si observamos la expresión matricial de la métrica es una matriz simétrica, no degenerada, y además siempre  $g(u, u) > 0$ .

El ejemplo anterior ilustra una situación donde el tensor métrico se obtiene como tensor métrico del *pull-back* del tensor métrico canónico de la métrica euclídea de  $\mathbb{R}^3$ . En este caso construimos la inmersión

$$f : \mathcal{S}^2 \rightarrow \mathbb{R}^3, \quad (\theta, \phi) \mapsto f(\theta, \phi) = (R \text{sen } \theta \cos \phi, R \text{sen } \theta \text{sen } \phi, R \cos \theta)$$

La métrica  $g$  es de hecho

$$g = f^* (dx \otimes dx + dy \otimes dy + dz \otimes dz).$$

El concepto del *pull-back* de un campo tensorial es mucho más extenso. Ponemos por completitud aquí una breve reseña.

**Definición 3.5.7** (Ver [12]). Sean  $\mathcal{M}$  y  $\mathcal{N}$  dos variedades diferenciales, sea  $f : \mathcal{N} \rightarrow \mathcal{M}$  y sea  $T$  un tensor de tipo  $(0, s)$  en  $\mathcal{M}$ . Entonces llamaremos tensor *pull-back* al tensor  $(0, s)$  en  $\mathcal{N}$  al tensor de tipo:

$$f^* T(v^1, \dots, v^s) = T_{f(p)}(df_{f(p)}(v^1), \dots, df_{f(p)}(v^s)) \quad (3.26)$$

Que será un campo tensorial para todo  $v \in T_p \mathcal{N}$ .

### 3.6. Conexiones

En esta sección vamos a introducir el concepto de conexión afín. Primero mostraremos el concepto de derivada covariante en una variedad diferenciable para posteriormente mostrar la idea de conexión afín y algunas ideas que nacen de la noción de conexión afín, como pueden ser los símbolos de Christoffel, transporte paralelo y la curva geodésica. Finalizaremos esta sección introduciendo el concepto de conexión de Levi-Civita y demostrando algunos resultados derivados de esta conexión.

**Definición 3.6.1** (Ver [5]). Sea  $\mathcal{M}$  una variedad diferenciable. Una **conexión afín**  $\nabla$  en  $\mathcal{M}$  es una aplicación:

$$\begin{aligned} \nabla : \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) &\rightarrow \mathfrak{X}(\mathcal{M}) \\ (X, Y) &\rightarrow \nabla_X Y \end{aligned}$$

Que debe de cumplir las siguientes propiedades para todo  $X, Y, Z \in \mathfrak{X}(\mathcal{M})$  y  $f, g \in \mathcal{C}^\infty(\mathcal{M})$ :

1.  $\nabla_{fX+gY}Z = f\nabla_X Z + g\nabla_Y Z$ , es decir es  $\mathcal{C}^\infty(\mathcal{M})$ -lineal respecto de  $X$
2.  $\nabla_X(Y + Z) = \nabla_X Y + \nabla_X Z$ , es decir es  $\mathbb{R}$ -lineal respecto de  $Y+Z$ .
3.  $\nabla_X(fY) = f\nabla_X Y + X(f)Y$

El concepto de conexión afín,  $\nabla_X Y$ , también se puede ver como la derivada covariante de  $Y$  respecto de  $X$  para la conexión  $\nabla$ .

**Definición 3.6.2** (Ver [5]). Sean  $\mathcal{M}$  una variedad diferenciable, sea  $\alpha : [-\varepsilon, \varepsilon] \rightarrow \mathcal{M}$  una curva sobre  $\mathcal{M}$  y  $V$  un campo vectorial definido sobre la curva  $\alpha$  y tangente a  $\mathcal{M}$ . Entonces, definiremos la **derivada covariante** de  $V$ , y lo denotaremos como  $\frac{DV}{dt}(t)$ , a la proyección ortogonal de  $\frac{dV}{dt}(t)$ .

**Proposición 3.6.3** (Ver [5]). Sea  $\mathcal{M}$  una variedad diferenciable y sea  $\nabla$  una **conexión afín**. Entonces existe una única función que asocia a un campo vectorial  $X$  sobre la curva diferenciable  $c : I \subset \mathbb{R} \rightarrow \mathcal{M}$  sobre otro campo vectorial  $\frac{DX}{dt}(t)$ , la derivada covariante de  $V$  sobre  $c$ , tal que:

1.  $\frac{D}{dt}(X + Y) = \frac{DX}{dt}(t) + \frac{DY}{dt}(t)$
2.  $\frac{D}{dt}(fX) = \frac{df}{dt}V + f\frac{DV}{dt}$
3. Si  $X$  está inducido por otro campo vectorial  $Y \in \mathfrak{X}\mathcal{M}$ , es decir,  $X(t)=Y(c(t))$  entonces  $\frac{DX}{dt}(t) = \nabla_{\frac{dc}{dt}} Y$

Entonces si tomamos una carta de  $\mathcal{M}$ ,  $v : U \subset \mathbb{R}^n \rightarrow \mathcal{M}$  y expresamos  $X$  e  $Y$  respecto de dicha carta:

$$\begin{aligned} X &= x^i \frac{\partial}{\partial v^i} \\ Y &= y^j \frac{\partial}{\partial v^j} \end{aligned} \quad (3.27)$$

Por lo tanto podemos expresar  $\nabla_X Y$  respecto a la carta  $v$  como:

$$\nabla_X Y = x^i \nabla_{\frac{\partial}{\partial v^i}} \left( y^j \frac{\partial}{\partial v^j} \right) = \left( x^i \cdot y^j \nabla_{\frac{\partial}{\partial v^i}} \frac{\partial}{\partial v^j} \right) + \left( x^i \frac{\partial y^j}{\partial v^i} \frac{\partial}{\partial v^j} \right) \quad (3.28)$$

Y si reescribimos  $\nabla_{\frac{\partial}{\partial v^i}} \frac{\partial}{\partial v^j} = \Gamma_{ij}^k \frac{\partial}{\partial v^k}$  entonces obtenemos:

$$\nabla_X Y = \left( x^i \cdot y^j \Gamma_{ij}^k + X y^k \right) \frac{\partial}{\partial v^k} \quad (3.29)$$

**Definición 3.6.4** (Ver [12]). A  $\Gamma_{ij}^k$  lo llamaremos **símbolo de Christoffel** para una parametrización  $v$ , es una función diferenciable de  $v(U) \rightarrow \mathbb{R}$ . No será un tensor debido a que no obedece las reglas de cambio de coordenadas, como se puede ver en [8]. Además, para una variedad semi-riemanniana  $(\mathcal{M}, g)$  existirán  $\dim(\mathcal{M})^3$  símbolos de Christoffel.

**Definición 3.6.5** (Ver [8]). Un campo vectorial,  $X$ , definido sobre una curva  $c : I \rightarrow \mathcal{M}$  se dice que es **paralelo en  $c$**  si:

$$\frac{DX}{dt}(t) = 0 \quad \forall t \in I \quad (3.30)$$

Si se escribe respecto de una carta,  $v : U \subset \mathbb{R}^n \rightarrow \mathcal{M}$  será:

$$\dot{X}^i + \Gamma_{jk}^i \dot{v}^j X^k = 0 \quad \forall i = 1, \dots, n \quad (3.31)$$

Que es un sistema de ecuaciones diferenciales lineales de primer orden para las componentes del campo  $X$ .

Ahora vamos a generalizar la idea de "línea recta" de la geometría euclídea a un tipo de curva en la geometría semi-Riemanniana y Riemanniana, **curva geodésica**.

**Definición 3.6.6.** (Ver [8]) Diremos que la curva  $c$  es una **geodésica** de la conexión  $\nabla$  si  $\dot{c}$  es paralelo en  $c$ :

$$\frac{D\dot{c}}{dt}(t) = 0 \quad \forall t \in I \quad (3.32)$$

Esta definición no es la única forma de obtener una curva geodésica. Se pueden obtener mediante las ecuaciones de Euler-Lagrange, pero no será el objetivo de este trabajo. Si se quiere profundizar en este tema se pueden ver los apuntes del profesor Fernando Chamizo [3].

Las ecuaciones de las curvas geodésicas respecto de una carta,  $\{U, v\}$ , serán:

$$\ddot{v}^i + (\Gamma_{jk}^i \dot{v}^j \dot{v}^k) = 0 \quad \forall i = 1, \dots, n \quad (3.33)$$



Es decir, un sistema de ecuaciones diferenciales de segundo grado no lineales. Entonces, por el teorema de Picard dados un punto,  $p \in \mathcal{M}$ , y un vector,  $v \in T_p\mathcal{M}$ , existirá una única geodésica,  $c : I \rightarrow \mathcal{M}$  tal que  $0 \in I$ , que satisfaga que  $c(0) = p$  y  $\dot{c}(0) = v$ .

**Definición 3.6.7** (Ver [8]). La torsión de una conexión  $\nabla$  en  $\mathcal{M}$  es una aplicación tal que:

$$\begin{aligned} T : \mathfrak{X}\mathcal{M} \times \mathfrak{X}\mathcal{M} &\rightarrow \mathfrak{X}\mathcal{M} \\ (X, Y) &\rightarrow \nabla_X Y - \nabla_Y X - [X, Y] \end{aligned}$$

Diremos que una conexión es **libre de torsión** o simétrica si  $T = 0$ , es decir si y solo si  $\nabla_X Y - \nabla_Y X = [X, Y]$ . Entonces, la torsión en coordenadas respecto de una carta será:

$$T = (\Gamma_{j,k}^i - \Gamma_{k,j}^i) dx^j \otimes dy^k \otimes \frac{\partial}{\partial x^i} \quad (3.34)$$

Que es un tensor de tipo  $(2,1)$ , es decir  $T \in \mathcal{T}^{(2,1)}(V, V^*)$ . Entonces, observando la ecuación (3.34) si una conexión es libre de torsión entonces debe de ocurrir que todos los términos sean nulos, por lo tanto:

$$\Gamma_{ij}^k = \Gamma_{ji}^k \quad \forall i, j, k = 1, \dots, n$$

### 3.6.1. Conexión de Levi-Civita

En el caso de las variedades semi-riemannianas existe un tipo específico de conexión, la **conexión de Levi-Civita**. En este apartado mostraremos dicha conexión y la fórmula de Koszul, la cual demostraremos que define una conexión de Levi-Civita. Finalizaremos con algunos resultados generales sobre las geodésicas derivadas de la conexión de Levi-Civita.

**Definición 3.6.8** (Ver [5]). Sea  $(\mathcal{M}, g)$  una variedad semi-riemanniana y sea  $\nabla$  una conexión en dicha variedad. Diremos que la conexión  $\nabla$  es **compatible** con la métrica  $g$  si y solamente si:

$$X \cdot g(Y, Z) = g(\nabla_X Y, Z) + g(Y, \nabla_X Z) \quad \forall X, Y, Z \in \mathfrak{X}\mathcal{M} \quad (3.35)$$

Para poder realizar la demostración del teorema siguiente necesitamos introducir la siguiente proposición técnica.

**Proposición 3.6.9** (Ver [5]). Sea  $(\mathcal{M}, g)$  una variedad riemanniana. Dado un  $V \in \mathfrak{X}(\mathcal{M})$  sea  $V^\flat$  la uno forma en  $\mathcal{M}$  tal que:

$$V^\flat(X) = g(V, X) \quad \forall X \in \mathfrak{X}\mathcal{M}$$

Entonces, la aplicación  $V \rightarrow V^\flat$  es un isomorfismo lineal de clase  $C^\infty$  de  $\mathfrak{X}(\mathcal{M})$

**Teorema 3.6.10** (Ver [12]). Sea  $(\mathcal{M}, g)$  una variedad semi-riemanniana entonces existe una única conexión afín  $\nabla$  en  $\mathcal{M}$  que satisface las siguientes condiciones:

- $\nabla$  es libre de torsión.
- $\nabla$  es compatible con la métrica semi-riemannianna.

A esta conexión se le conoce como **conexión de Levi-Civita**

*Demostración.* Supongamos que existe dicha conexión,  $\nabla$ , y que cumple que es compatible con la métrica, entonces:

1.  $Xg(Y, Z) = g(\nabla_X Y, Z) + g(Y, \nabla_X Z)$
2.  $Yg(Z, X) = g(\nabla_Y Z, X) + g(Z, \nabla_Y X)$
3.  $Zg(X, Y) = g(\nabla_Z X, Y) + g(X, \nabla_Z Y)$

Entonces sumamos 1 y 2 y restamos 3:

$$Xg(Y, Z) + Yg(Z, X) - Zg(X, Y) = g(\nabla_X Y, Z) + g(Y, \nabla_X Z) + g(Z, \nabla_Y X) + g(\nabla_Y Z, X) - g(\nabla_Z X, Y) - g(X, \nabla_Z Y) \quad (3.36)$$

Como  $g$  es simétrica:

$$Xg(Y, Z) + Yg(Z, X) - Zg(X, Y) = g(\nabla_X Y, Z) + g(\nabla_X Z, Y) + 2g(\nabla_Y Z, X) - g(\nabla_Z X, Y) - g(\nabla_Z Y, X) \quad (3.37)$$

Aprovechando que  $\nabla$  es libre de torsión :

$$Xg(Y, Z) + Yg(Z, X) - Zg(X, Y) = g([X, Y], Z) + g([Y, Z], X) + g([X, Y], Z) + 2g(Z, \nabla_Y X) \quad (3.38)$$

Entonces a esta igualdad se le llama **fórmula de Koszul**:

$$g(Z, \nabla_Y X) = \frac{1}{2} \left( Xg(Y, Z) + Yg(Z, X) - Zg(X, Y) - g([X, Z], Y) - g([Y, Z], X) - g([X, Y], Z) \right) \quad (3.39)$$

Por lo tanto como  $(\mathcal{M}, g)$  es una variedad semi-riemanniana entonces  $g$  es no degenerada y  $Z$  es un campo vectorial cualquiera entonces esta fórmula determina la conexión  $\nabla_X Y$ . Por ende, si la conexión de Levi-Civita existe debe de ser única.

Nos queda por demostrar que existe dicha conexión. Para ello vamos a definir la conexión en base a la fórmula de Koszul y vamos a comprobar que cumple todas las propiedades de una

conexión afín además de que es simétrica y compatible con la métrica. Para probar que cumple las propiedades de una conexión afín tendremos los campos vectoriales  $X, Y, Z, V \in \mathfrak{X}(\mathcal{M})$ . Satisface las siguientes propiedades para todo  $X, Y, Z \in \mathfrak{X}(\mathcal{M})$  y  $f, g \in C^\infty(\mathcal{M})$ :

$$\begin{aligned}
g(\nabla_{fX+hY}Z, V) &= \frac{1}{2} \left( (fX + hY)g(Y, Z) + Zg((fX + hY), V) \right. \\
&\quad \left. - Vg((fX + hY), Z) - g([(fX + hY), V], Z) \right. \\
&\quad \left. g([Z, V], (fX + hY)) - g([(fX + hY), Z], V) \right) \\
&= \frac{1}{2} \left( fXg(Y, Z) + hYg(Y, Z) + Zg(fX, V) + Zg(hY, V) - Vg(fX, Z) \right. \\
&\quad \left. - Vg(hY, Z) - g([fX, V], Z) - g([hY, V], Z) - g([Z, V], hY) \right. \\
&\quad \left. - g([Z, V], fX) - g([fX, Z], V) - g([hY, Z], V) \right) \tag{3.40} \\
&= \frac{1}{2} \left( fXg(Y, Z) + Zg(fX, V) - Vg(fX, Z) - g([fX, V], Z) \right. \\
&\quad \left. - g([Z, V], fX) - g([fX, Z], V) \right) + \frac{1}{2} \left( hYg(Y, Z) + Zg(hY, V) \right. \\
&\quad \left. - Vg(hY, Z) - g([hY, V], Z) - g([Z, V], hY) - g([hY, Z], V) \right) \\
&= fg(\nabla_X Z, V) + hg(\nabla_Y Z, V)
\end{aligned}$$

$$\begin{aligned}
g(\nabla_X(Y + Z), V) &= \frac{1}{2} \left( Xg(Y + Z, V) + (Y + Z)g(V, X) - Vg(X, Y + Z) \right. \\
&\quad \left. - g([X, V], Y + Z) - g([Y + Z, V], X) - g([X, Y + Z], V) \right) \\
&= \frac{1}{2} \left( Xg(Z, V) + Xg(Y, V) + Yg(V, X) + Zg(V, X) \right. \\
&\quad \left. - Vg(X, Y) - Vg(X, Z) - g([X, Z], V) - g([X, Y], V) \right. \\
&\quad \left. - g([Z, V], X) - g([Y, V], X) - g([X, V], Y) - g([X, V], Z) \right) \tag{3.41} \\
&= \frac{1}{2} \left( Xg(Z, V) + Zg(V, X) - Vg(X, Z) - g([X, Z], V) - g([Z, V], X) \right. \\
&\quad \left. - g([X, V], Z) \right) + \frac{1}{2} \left( Xg(Y, V) + Yg(V, X) - Vg(X, Y) \right. \\
&\quad \left. - g([X, Y], V) - g([Y, V], X) - g([X, V], Z) \right) \\
&= g(\nabla_X Y, V) + g(\nabla_X Z, V)
\end{aligned}$$

$$\begin{aligned}
g(\nabla_X(fY), Z) &= \frac{1}{2} \left( Xg(fY, Z) + fYg(X, Z) - Zg(X, fY) + g(X, [fY, Z], Z) \right. \\
&\quad \left. + g(fY, [Z, X]) + g(X, [Z, fY]) \right) \\
&= g(f\nabla_X Y, Z) + g(X(f)Y, Z)
\end{aligned} \tag{3.42}$$

Para probar que la conexión es libre de torsión vamos a usar también la fórmula de Koszul. Sean  $X, Y, Z \in \mathfrak{X}(\mathcal{M})$ :

$$\begin{aligned}
g(\nabla_X Y - \nabla_Y X, Z) &= g(\nabla_X Y, Z) - g(\nabla_Y X, Z) \\
&= \frac{1}{2} \left( Xg(Y, Z) + Yg(Z, X) - Zg(X, Y) \right. \\
&\quad \left. - g([X, Y], Z) - g([Y, Z], X) - g([X, Z], Y) \right) \\
&\quad - \frac{1}{2} \left( Yg(X, Z) + Xg(Z, Y) - Zg(Y, X) \right. \\
&\quad \left. - g([Y, X], Z) - g([X, Z], Y) - g([Y, Z], X) \right) \\
&= g([X, Y], Z)
\end{aligned} \tag{3.43}$$

Ahora falta demostrar que esta conexión es compatible con la métrica:

$$\begin{aligned}
g(\nabla_X Y, Z) + g(Y, \nabla_X Z) &= \frac{1}{2} \left( Xg(Y, Z) + Yg(Z, X) - Zg(X, Y) \right. \\
&\quad \left. - g([X, Y], Z) - g([Y, Z], X) - g([X, Z], Y) \right) + \\
&\quad \frac{1}{2} \left( Xg(Z, Y) + Zg(Y, X) - Yg(X, Z) \right. \\
&\quad \left. - g([X, Z], Y) - g([Z, Y], X) - g([X, Y], Z) \right) \\
&= X \cdot g(Y, Z)
\end{aligned} \tag{3.44}$$

Por lo tanto hemos demostrado que la fórmula de Koszul determina una conexión libre de torsión y compatible con la métrica, además, es única.  $\square$

A partir de la fórmula de Koszul se puede obtener la siguiente igualdad para poder obtener la expresión de los símbolos de Christoffel de una forma más simple. Tomamos una coordenadas

locales  $(x^i)$  entonces tenemos que:

$$\begin{aligned} \left[ \frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j} \right] &= 0 \\ g \left( \frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j} \right) &= g_{ij} \end{aligned}$$

Pero para simplificar la notación vamos a denotar la derivada de  $g\left(\frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j}\right)$  respecto de  $x^k$  se denotará como  $g_{ij,k}$ , es decir:

$$g_{ij,k} := \frac{\partial}{\partial x^k} g \left( \frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j} \right) \quad (3.45)$$

También para denotar el elemento  $i,j$  de la expresión matricial de  $g$  inversa,  $g^{-1}$  se expresa de la siguiente forma:  $g^{ij}$ .

$$\begin{aligned} 2g \left( \nabla_{\frac{\partial}{\partial x^k}} \frac{\partial}{\partial x^l}, \frac{\partial}{\partial x^i} \right) &= \frac{\partial}{\partial x^j} \cdot g_{kl} + \frac{\partial}{\partial x^k} \cdot g_{jl} - \frac{\partial}{\partial x^l} \cdot g_{kj} \\ \Leftrightarrow g \left( \Gamma_{jk}^i \frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^l} \right) &= \frac{1}{2} \left( \frac{\partial g_{kl}}{\partial x^j} + \frac{\partial g_{jl}}{\partial x^k} - \frac{\partial g_{kj}}{\partial x^l} \right) \\ \Gamma_{jk}^i g_{il} &= \frac{1}{2} \left( \frac{\partial g_{kl}}{\partial x^j} + \frac{\partial g_{jl}}{\partial x^k} - \frac{\partial g_{kj}}{\partial x^l} \right) \end{aligned} \quad (3.46)$$

Una vez se han mostrados estas dos simplificaciones vamos a mostrar la forma simplificada de calcular los símbolos de Christoffel es:

$$\Gamma_{jk}^i = \frac{g^{il}}{2} \left( \frac{\partial g_{kl}}{\partial x^j} + \frac{\partial g_{jl}}{\partial x^k} - \frac{\partial g_{kj}}{\partial x^l} \right) \quad (3.47)$$

Otro resultado que se obtiene de la conexión de Levi-Civita es:

**Proposición 3.6.11.** *Sea  $\mathcal{M}$  una variedad semi-riemanniana con métrica  $g$  y sea  $\vec{v}(\tau)$  el vector tangente a una geodésica entonces  $g(\vec{v}(\tau), \vec{v}(\tau))$  es constante.*

*Demostración.* Sea  $\gamma(\lambda)$  una geodésica. Para simplificar la notación denotaremos como

$$\dot{\gamma}(t) = \frac{d\gamma}{d\lambda}.$$

$$\begin{aligned} \frac{d}{d\lambda} g(\dot{\gamma}(t), \dot{\gamma}(t)) &= \frac{d}{d\lambda} g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t)) \\ &= \dot{\gamma}(t) \left( g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t)) \right) \\ &= 2g_{\gamma(t)} \left( \nabla_{\dot{\gamma}(t)} \dot{\gamma}(t), \dot{\gamma}(t) \right) \end{aligned} \quad (3.48)$$

Como  $\gamma$  es una geodésica obtenemos que  $\nabla_{\dot{\gamma}(t)}\dot{\gamma}(t) = 0$  y por lo tanto:

$$\frac{d}{d\lambda}g(\dot{\gamma}(t), \dot{\gamma}(t)) = 2g_{\gamma(t)}(0, \dot{\gamma}(t)) \quad (3.49)$$

Y como  $g$  es una métrica semi-riemanniana  $g(0, v) = 0$  y por lo tanto:

$$\frac{d}{d\lambda}g(\dot{\gamma}(t), \dot{\gamma}(t)) = 0 \quad (3.50)$$

En consecuencia obtenemos que:

$$g(\dot{\gamma}(t), \dot{\gamma}(t)) = k \in \mathbb{R} \quad (3.51)$$

Lo que viene a decir es que los vectores de una geodésica no ven su tipo alterado.

### 3.7. Curvatura

Como previamente hemos mencionado, existen propiedades de las variedades que únicamente depende de ellas mismas. Una de estas es la curvatura, la noción de curvatura fue introducida por Gauss en su artículo de 1827 "Investigación general de superficies curvadas" en superficies embebidas en  $\mathbb{R}^3$  y fue generalizado por Riemann en 1860.

La idea de curvatura se puede ver desde diferentes ángulos: el ratio de desviación de las geodésicas respecto de una línea recta, la suma de los ángulos de un triángulo geodésicas o el grado de no conmutatividad de las derivadas covariantes sobre diferentes campos vectoriales.

En esta sección comenzaremos hablando de la idea del tensor curvatura de una conexión afín partiendo de la curvatura de Gauss, y demostraremos algunas de las propiedades de esta. Posteriormente introduciremos el tensor curvatura y la noción de curvatura seccional. Culminaremos esta sección con la curvatura de Ricci y la curvatura escalar.

Como punto de partida para hablar de la curvatura comenzaremos recordando la noción de curvatura de Gauss que se vio en la asignatura de Geometría diferencial y topología. La curvatura de Gauss en cada punto era el producto de las curvaturas normales máxima y mínimas en dicho punto. Además, como enuncia el *Teorema Egregium de Gauss* esta curvatura será invariante bajo isometrías locales, por lo que la curvatura de Gauss es una buena forma de medir la curvatura de una superficie. La curvatura de una variedad Riemanniana de dimensión 2 cuantifica la diferencia de la superficie con el plano midiendo los ángulos de un triángulo geodésico<sup>4</sup> en dicha variedad. Es decir, mide la variación en función del área que sufren los vectores en transportes paralelos cíclicos en un triángulo geodésico.

---

<sup>4</sup>Triángulo cuyos vértices son todos curvas geodésicas.

Esta noción de curvatura no involucra ninguna consideración de que es una variedad de dimensión dos, esto nos permitirá encontrar un tensor. Este tensor es el tensor de curvatura de Riemann que lo generaliza a variedades de dimensión mayor que 2, pero en vez de un triángulo usando un paralelogramo.

**Definición 3.7.1** (Ver [8]). Sea  $\mathcal{M}$  una variedad semi-Riemanniana y sea  $\nabla$  la conexión de Levi-Civita. La **curvatura**,  $R$ , de la conexión  $\nabla$  es una aplicación que para cada terna de campos vectoriales ,  $X, Y, Z \in \mathfrak{X}(\mathcal{M})$ , les asocia otro campo vectorial de la forma:

$$\begin{aligned} R : \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) &\rightarrow \mathfrak{X}(\mathcal{M}) \\ (X, Y, Z) &\rightarrow R(X, Y)Z = \nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z - \nabla_{[X, Y]} Z \end{aligned} \quad (3.52)$$

Es un campo tensorial (1,3), y se le llama **Tensor Curvatura Riemann** de  $\mathcal{M}$ .

Para poder manejar de una forma más simple la curvatura vamos a llevarlo al mundo de las cartas. Sea  $(U, x)$  una carta de  $\mathcal{M}$ . Entonces, la curvatura  $R$  expresada respecto de la carta  $(U, x)$  obtenemos que:

$$R = R^l_{ijk} \cdot dx^i \otimes dx^j \otimes dx^k \otimes \frac{\partial}{\partial x^l} \quad (3.53)$$

Donde cada coeficiente  $R^l_{ijk}$  es la  $l$ -ésima coordenada del campo vectorial  $R\left(\frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j}\right)\frac{\partial}{\partial x^k}$ , esto es:

$$R\left(\frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j}\right)\frac{\partial}{\partial x^k} = \nabla_{\frac{\partial}{\partial x^i}} \nabla_{\frac{\partial}{\partial x^j}} \frac{\partial}{\partial x^k} - \nabla_{\frac{\partial}{\partial x^j}} \nabla_{\frac{\partial}{\partial x^i}} \frac{\partial}{\partial x^k} - \nabla_{[\frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j}]} \frac{\partial}{\partial x^k} \quad (3.54)$$

Entonces como  $[\frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j}] = 0$

$$\begin{aligned} R\left(\frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j}\right)\frac{\partial}{\partial x^k} &= \nabla_{\frac{\partial}{\partial x^i}} \nabla_{\frac{\partial}{\partial x^j}} \frac{\partial}{\partial x^k} - \nabla_{\frac{\partial}{\partial x^j}} \nabla_{\frac{\partial}{\partial x^i}} \frac{\partial}{\partial x^k} \\ &\text{Por la definición de símbolo de Christoffel:} \\ &= \nabla_{\frac{\partial}{\partial x^i}} \left( \Gamma^m_{jk} \frac{\partial}{\partial x^m} \right) - \nabla_{\frac{\partial}{\partial x^j}} \left( \Gamma^m_{ik} \frac{\partial}{\partial x^m} \right) \\ &= \left( \frac{\partial}{\partial x^i} \Gamma^m_{jk} - \frac{\partial}{\partial x^j} \Gamma^m_{ik} \right) \frac{\partial}{\partial x^m} + \left( \Gamma^m_{jk} \Gamma^l_{im} - \Gamma^m_{ik} \Gamma^l_{jm} \right) \frac{\partial}{\partial x^l} \\ &= \left( \frac{\partial \Gamma^l_{jk}}{\partial x^i} - \frac{\partial \Gamma^l_{ik}}{\partial x^j} + \Gamma^m_{jk} \Gamma^l_{im} - \Gamma^m_{ik} \Gamma^l_{jm} \right) \frac{\partial}{\partial x^l} \end{aligned} \quad (3.55)$$

Por lo tanto:

$$R^l_{ijk} = \frac{\partial \Gamma^l_{jk}}{\partial x^i} - \frac{\partial \Gamma^l_{ik}}{\partial x^j} + \Gamma^m_{jk} \Gamma^l_{im} - \Gamma^m_{ik} \Gamma^l_{jm} \quad (3.56)$$

**Proposición 3.7.2** (Ver [8]). Sean  $R$  un tensor de Curvatura de Riemann de  $\mathcal{M}$ ,  $f, g \in D(\mathcal{M})$  y  $Z, W \in \mathfrak{X}(\mathcal{M})$ . Entonces, el tensor curvatura de Riemann cumplirá que:

1.  $R$  es bilineal. Demostración.

$$\begin{aligned}
R(fX + gY, Z)W &= \nabla_{fX+gY} \nabla_Z W - \nabla_Z \nabla_{fX+gY} Z - \nabla_{[fX+gY, Z]} W \\
&= f \nabla_X \nabla_Z W + g \nabla_Y \nabla_Z W - \nabla_Z (f \nabla_X + g \nabla_Y) Z \\
&\quad - \nabla_{f[X, Z]} W - \nabla_{g[Y, Z]} W \\
&= f \nabla_X \nabla_Z W + g \nabla_Y \nabla_Z W - \nabla_Z f \nabla_X Z + \nabla_Z g \nabla_Y Z \\
&\quad - \nabla_{f[X, Z]} W - \nabla_{g[Y, Z]} W \\
&= (f \nabla_X \nabla_Z W - \nabla_Z f \nabla_X Z - \nabla_{f[X, Z]} W) \\
&\quad + (g \nabla_Y \nabla_Z W + \nabla_Z g \nabla_Y Z - \nabla_{g[Y, Z]} W) \\
&= fR(X, Z)W + gR(Y, Z) \quad \square
\end{aligned} \tag{3.57}$$

2.  $R$  es un operador lineal para todo par de campos vectoriales. Demostración.

a)  $R(X, Y)(Z + W) = R(X, Y)(Z) + R(X, Y)(W)$

$$\begin{aligned}
R(X, Y)(Z + W) &= \nabla_X \nabla_Y (Z + W) - \nabla_Y \nabla_X (Z + W) - \nabla_{[X, Y]} (Z + W) \\
&= \nabla_X (\nabla_Y Z + \nabla_Y W) - \nabla_Y (\nabla_X Z + \nabla_X W) \\
&\quad - (\nabla_{[X, Y]}(Z) + \nabla_{[X, Y]}(W)) \\
&= \nabla_X \nabla_Y W + \nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z - \nabla_Y \nabla_X W \\
&\quad - \nabla_{[X, Y]}(Z) + \nabla_{[X, Y]}(W) \\
&= (\nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z - \nabla_{[X, Y]}(Z)) \\
&\quad + (\nabla_X \nabla_Y W - \nabla_Y \nabla_X W - \nabla_{[X, Y]}(W)) \\
&= R(X, Y)(Z) + R(X, Y)(W)
\end{aligned} \tag{3.58}$$

□

La linealidad de la segunda componente se demuestra de forma análoga.

b)  $R(X, Y)fZ = fR(X, Y)Z$  Demostración.

$$\begin{aligned}
R(X, Y)fZ &= \nabla_X \nabla_Y fZ - \nabla_Y \nabla_X fZ - \nabla_{[X, Y]} fZ \\
&= \nabla_X (f \nabla_Y Z + Y(f)Z) - \nabla_Y (f \nabla_X Z + X(f)Z) \\
&\quad - f \nabla_{[X, Y]} Z - [X, Y]fZ \\
&= \nabla_X f \nabla_Y Z + \nabla_X Y(f)Z - \nabla_Y f \nabla_X Z - \nabla_Y X(f)Z \\
&\quad - f \nabla_{[X, Y]} Z - [X, Y]fZ \\
&= (\nabla_X f \nabla_Y Z - \nabla_Y f \nabla_X Z - f \nabla_{[X, Y]} Z) \\
&\quad + \nabla_X (Y(f)Z) - \nabla_Y (X(f)Z) - [X, Y]fZ \\
&= fR(X, Y) + 0 \square
\end{aligned} \tag{3.59}$$



**Proposición 3.7.3** (Ver [8]). (*Identidad de Bianchi*) Sea  $R$  un campo tensorial de Riemann y sean  $X, Y, Z \in \mathfrak{X}(\mathcal{M})$  Entonces:

$$R(X, Y)Z + R(Y, Z)X + R(Z, X)Y = 0 \quad (3.60)$$

*Demostración.*

$$\begin{aligned} R(X, Y)Z + R(Y, Z)X + R(Z, X)Y &= \nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z - \nabla_{[X, Y]} Z \\ &\quad + \nabla_Y \nabla_Z X - \nabla_Z \nabla_Y X - \nabla_{[Y, Z]} X \\ &\quad + \nabla_Z \nabla_X Y - \nabla_X \nabla_Z Y - \nabla_{[Z, X]} Y \end{aligned}$$

Como la conexión es simétrica

$$\begin{aligned} &= \nabla_X [Z, Y] + \nabla_Y [X, Z] + \nabla_Z [Y, X] \\ &\quad - \nabla_{[X, Y]} Z - \nabla_{[Y, Z]} X - \nabla_{[Z, X]} Y \\ &= [Y, [X, Z]] + [Z, [Y, X]] + [X, [Z, Y]] \end{aligned} \quad (3.61)$$

Por la identidad de Jacobi

$$= 0 \quad \square$$

El tensor curvatura de Riemann tendrá  $n^4$  componentes para una variedad  $n$ -dimensional, pero si recordamos en el caso de una variedad que habita en  $\mathbb{R}^3$  solamente necesitamos la curvatura de Gauss. Esto nos indica que hay componentes que son redundantes en el tensor curvatura de Riemann, para ello vamos a definir el tensor curvatura de tipo (0,4), de tal forma que no se pierda información y podamos ver estas simetrías.

**Definición 3.7.4** (Ver [8]). Sea  $(\mathcal{M}, g)$  una variedad Riemanniana y sea  $\nabla$  la conexión de Levi-Civita. Entonces, se define el tensor curvatura como:

$$R(X, Y, Z, W) := g(R(X, Y)Z, W) \quad (3.62)$$

Entonces, si tomamos una carta  $(U, x)$  de  $\mathcal{M}$  se puede escribir el tensor curvatura respecto de la carta como:

$$R(X, Y, Z, W) = (R_{ijkl} \cdot dx^i \otimes dx^j \otimes dx^k \otimes \frac{\partial}{\partial x^l})(X, Y, Z, W) \quad (3.63)$$

Cada componente del tensor curvatura,  $R_{ijkl}$  será:

$$\begin{aligned} R_{ijkl} &= g \left( R \left( \frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j} \right) \frac{\partial}{\partial x^k}, \frac{\partial}{\partial x^l} \right) \\ &= g \left( R_{ijk}^m \frac{\partial}{\partial x^m}, \frac{\partial}{\partial x^l} \right) \\ &= R_{ijk}^m \cdot g_{ml} \end{aligned} \quad (3.64)$$

**Proposición 3.7.5** (Ver [8]). Sean  $(\mathcal{M}, g)$  una variedad Riemanniana y sea  $(U, x)$  una carta de la variedad Riemanniana. El tensor curvatura se puede calcular en coordenadas locales mediante la siguiente forma:

$$R_{ijkl} = \frac{1}{2} \left( \frac{\partial^2 g_{jl}}{\partial x^i \partial x^k} - \frac{\partial^2 g_{il}}{\partial x^j \partial x^k} - \frac{\partial^2 g_{jk}}{\partial x^i \partial x^l} + \frac{\partial^2 g_{ik}}{\partial x^j \partial x^l} \right) \quad (3.65)$$

*Demostración.* Primero vamos a expresar el tensor curvatura respecto de la carta, (3.64). Si sustituimos los componentes del tensor de Riemann, (3.56), y obtenemos que:

$$R_{ijkl} = \left( \frac{\partial \Gamma_{jk}^l}{\partial x^i} - \frac{\partial \Gamma_{ik}^l}{\partial x^j} + \Gamma_{jk}^w \Gamma_{iw}^l - \Gamma_{ik}^w \Gamma_{jw}^l \right) \cdot g_{wl} \quad (3.66)$$

Además, expresamos los símbolos de Christoffel respecto de dicha carta, mediante el uso de la fórmula de Koszul, y la expresión queda de la siguiente forma:

$$\begin{aligned} R_{ijkl} = & \left( \frac{\partial \left( \frac{g^{lm}}{2} \left( \frac{\partial g_{mj}}{\partial x^k} + \frac{\partial g_{mk}}{\partial x^j} - \frac{\partial g_{jk}}{\partial x^m} \right) \right)}{\partial x^i} - \frac{\partial \left( \frac{g^{lm}}{2} \left( \frac{\partial g_{mi}}{\partial x^k} + \frac{\partial g_{mk}}{\partial x^i} - \frac{\partial g_{ik}}{\partial x^m} \right) \right)}{\partial x^j} \right. \\ & + \left( \frac{g^{wm}}{2} \left( \frac{\partial g_{mj}}{\partial x^k} + \frac{\partial g_{mk}}{\partial x^j} - \frac{\partial g_{jk}}{\partial x^m} \right) \right) \left( \frac{g^{lm}}{2} \left( \frac{\partial g_{mi}}{\partial x^w} + \frac{\partial g_{mw}}{\partial x^i} - \frac{\partial g_{iw}}{\partial x^m} \right) \right) \\ & \left. - \left( \frac{g^{wm}}{2} \left( \frac{\partial g_{mi}}{\partial x^k} + \frac{\partial g_{mk}}{\partial x^i} - \frac{\partial g_{ik}}{\partial x^m} \right) \right) \left( \frac{g^{lm}}{2} \left( \frac{\partial g_{mj}}{\partial x^w} + \frac{\partial g_{mw}}{\partial x^j} - \frac{\partial g_{wj}}{\partial x^m} \right) \right) \right) \cdot g_{wl} \end{aligned} \quad (3.67)$$

Entonces simplificando la expresión anterior, obtenemos que:

$$R_{ijkl} = \frac{1}{2} \left( \frac{\partial^2 g_{jl}}{\partial x^i \partial x^k} - \frac{\partial^2 g_{il}}{\partial x^j \partial x^k} - \frac{\partial^2 g_{jk}}{\partial x^i \partial x^l} + \frac{\partial^2 g_{ik}}{\partial x^j \partial x^l} \right) \quad \square \quad (3.68)$$

Para observar las simetrías del tensor curvatura de Riemann vamos a ver la siguiente proposición.

**Proposición 3.7.6** (Ver [8]). Sean  $X, Y, Z, W \in \mathfrak{X}(\mathcal{M})$  y sea  $\nabla$  la conexión de Levi-Civita, entonces:

1.  $R(X, Y, Z, W) + R(Y, Z, X, W) + R(Z, X, Y, W) = 0$
2.  $R(X, Y, Z, W) = -R(Y, X, Z, W)$
3.  $R(X, Y, Z, W) = -R(X, Y, W, Z)$
4.  $R(X, Y, Z, W) = R(Z, W, X, Y)$

**Definición 3.7.7** (Ver [8]). Sea  $\Pi$  un subespacio vectorial de dimensión 2 del plano tangente a  $\mathcal{M}$  en el punto  $p$ . Se define la **curvatura seccional** de  $\Pi$  en  $p$ , y se denotará por  $K(\Pi)$ , como:

$$K(u, v) = -\frac{R(u, v, u, v)}{g(u, u)g(v, v) - g(u, v)^2} \quad (3.69)$$

Para toda base  $\{u, v\}$  de  $\Pi$ .

Se dirá que una variedad es **isotrópica** en un  $p$  si curvatura seccional es constante para todo subespacio bidimensional del espacio tangente en  $p$ . En el caso de que la variedad sea 2-dimensional cabe destacar que será isotrópico y en este caso la curvatura seccional será la **curvatura de Gauss**.

**Proposición 3.7.8** (Ver [8]). El tensor curvatura de Riemann en un punto  $p$  está únicamente determinado por los valores de las curvaturas seccionales.

Como hemos visto el tensor de Riemann tiene muchas componentes y muchas de ellas están relacionadas entre sí. Esto nos lleva a pensar que se puede definir un nuevo tensor partiendo de este con menos componentes y que mantenga la mayoría de la información. Este es el tensor de Ricci.

**Definición 3.7.9** (Ver [8]). El **Tensor de curvatura de Ricci** es un tensor  $(0,2)$  localmente definido como:

$$Ric(X, Y) = dx^k \left( R \left( \frac{\partial}{\partial x^k}, X \right) Y \right) \quad (3.70)$$

Localmente lo podemos escribir el tensor de la siguiente forma:

$$Ric = R_{ij} dx^i \otimes dx^j \quad (3.71)$$

Donde:

$$R_{ij} = Ricc \left( \frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j} \right) = dx^k \left( R \left( \frac{\partial}{\partial x^k}, \frac{\partial}{\partial x^i} \right) \frac{\partial}{\partial x^j} \right) = R_{kij}^k \quad (3.72)$$

**Proposición 3.7.10.** (Ver [11]) El tensor de Riemann y el tensor de Ricci proporcionan la misma información sobre la variedad si la dimensión de estas es menor o igual a 3

Por último vamos a definir la **curvatura escalar**, en la Relatividad General también se le conoce como escalar de Ricci. Esta idea surge de contraer los dos índices del tensor de Ricci con el tensor métrico obteniendo un tensor de tipo  $(0,0)$  que no dependerá de la carta usada, por ser un tensor de tipo  $(0,0)$ .

$$R = g^{ij} R_{ij} \quad (3.73)$$



# Capítulo 4

## Resultados

Con el objetivo de poder estudiar diferentes métricas se han desarrollado diferentes funciones de Python para calcular los símbolos de Christoffel, las curvas geodésicas y las componentes de los diferentes tensores de curvatura así como la curvatura escalar. Todo ello se aplicó en la métrica hiperbólica y en la métrica de Schwarzschild.

### 4.1. Códigos

Estos códigos están desarrollados en el lenguaje de programación Python utilizando principalmente las librerías de SciPy, para cálculo simbólico, y NumPy, para el manejo de las matrices.

En esta sección se mostrarán los códigos desarrollados usando algunas fórmulas que se muestran a lo largo del trabajo. Pero antes se explicará cómo se interacciona con estas funciones.

Para introducir la métrica que se quiere estudiar se debe de expresar de forma matricial. Para calcular la inversa de cualquier matriz se desarrolló una función que calculaba la matriz mediante la ecuación:

$$M^{-1} = \frac{Adj(M^T)}{det(M)} \quad (4.1)$$

Esto se hizo porque la librería de SciPy no trabaja bien con funciones y producía errores al calcular la inversa.

---

```
1 def inversa(g):
```

```

2      #Calculo el determinante
3      det = g.det()
4      #Calculo la transpuesta
5      g_transpuesta = g.transpose()
6      #La paso a un array de numpy
7      g_transpuesta = np.array(g_transpuesta)
8
9      #Hago una copia para poder
10     g_adjunta_de_la_transpuesta = np.array(g_transpuesta).copy()
11     for i in range(len(g_transpuesta)):
12         for j in range(len(g_transpuesta[i])):
13             matriz_copia = g_transpuesta.copy()
14             #Elimino las filas y las columnas
15             sub_matriz = np.delete(matriz_copia, axis = 0, obj=i)
16             sub_matriz = np.delete(sub_matriz, axis = 1, obj=j)
17             #Calculo el determinante de la submatriz
18             det_sub = sp.Matrix(sub_matriz).det()
19
20             g_adjunta_de_la_transpuesta[i][j] = (-1)**(i+j)* det_sub
21
22     g_adjunta_de_la_transpuesta = sp.Matrix(g_adjunta_de_la_transpuesta)
23     #Multiplico por el inverso del determinante
24     g_inv = sp.simplify( g_adjunta_de_la_transpuesta*(1/det))
25     #Lo paso a un array de numpy para poder acceder a los elementos
26     g_inv = np.array(g_inv)
27     return g_inv

```

---

#### 4.1.1. Código para calcular las geodésicas

Como se ha mencionado previamente para poder calcular los símbolos de Christoffel y las geodésicas se hicieron dos funciones en Python basándonos en las ecuaciones (3.47) y (3.33).

Primero se calculan sus derivadas covariantes de cada uno de los coeficientes de la matriz y son almacenadas en una matriz n-dimensional, donde n es las dimensión. Esto se ha hecho para así ahorrar tiempo a la hora de calcular los símbolos de Christoffel debido a que el tiempo que se invierte en calcular las derivadas es muy grande en comparación con la memoria que ocupa.

Una vez se han calculado las derivadas covariantes ya pueden se calculan las geodésicas para ello se utilizará la expresión (3.47). Por lo tanto las geodésicas serán almacenadas en una matriz

tridimensional donde el primer índice hará referencia a la variable superior y los dos siguientes harán referencia a las variables inferiores, de la siguiente forma:

$$\text{simbolo\_chirtoffel}[2][0][1] \equiv \Gamma_{t,x}^y$$

---

```

1 def simbolos_christoffel(g,variables):
2     #Calculo de la inversa
3     g_inv = inversa(g)
4
5     g = np.array(g)
6     #Calculo de las derivadas
7     #Este vector va a contener la matriz g derivada respecto de cada la variable
8     #Por ejemplo derivadas_variable[0] es la derivada de toda la matriz respecto de t
9     #     derivadas_variable[1] es la derivada de toda la matriz respecto de x
10    #Y derivadas_variable[1][0][0] es la derivad parcial de g_{0,0} respecto x
11    # y así sucesivamente
12    derivadas_variable = []
13
14    for variable in variables:
15        derivada_variable = [[0 for i in variables] for j in variables]
16        for i in range(len(variables)):
17            for j in range(len(variables)):
18                funcion = g[i][j]
19                #Para no hacer la derivada de un número
20                if type(funcion) in [ sp.core.mul.Mul ,sp.core.power.Pow,sp.core.add.Add]:
21                    derivada_variable[i][j] = funcion.diff(variable)
22        derivadas_variable.append(derivada_variable)
23
24    #Calculo de los simbolos de Christoffel
25    simbolos_chistoffel = [[[0 for k in variables] for j in variables] for i in variables]
26
27
28    for i in range(len(variables)):
29        for j in range(len(variables)):
30            for k in range(len(variables)):
31                simbolo = 0
32                #Indice libre l
33                for l in range(len(variables)):
34                    if g_inv[i][l] != 0:

```

```

35         simbolo =simbolo+ g_inv[i][l]*(derivadas_variable[j][k][l]
36             derivadas_variable[k][j][l]-derivadas_variable[l][j][k])
37     simbolos_chistoffel[i][j][k] = sp.simplify(1/2*simbolo)
38
39     return simbolos_chistoffel

```

---

Cuando ya se tienen calculados todos los símbolos de Christoffel se pueden calcular las geodésicas según la expresión (3.33), como se puede ver en el siguiente código.

---

```

1 def geodesicas(simbolos_christoffel, variables, variables_derivadas):
2     geodesicas = [i.diff(tau) for i in variables_derivadas]
3     for variable in range(len(variables)):
4         for i in range(len(variables)):
5             for j in range(len(variables)):
6                 geodesicas[variable] = geodesicas[variable] +sp.simplify(
7                     simbolos_chistoffel[variable][i][j]*variables_derivadas[i]*
8                     variables_derivadas[j])
9         geodesicas[variable] = sp.Eq(geodesicas[variable],0)
10    return geodesicas

```

---

Una vez se tiene la expresión de las geodésicas se puede intentar resolver con la función de *SciPy* para resolver sistemas de ecuaciones. Usar esta función no es posible en la mayoría de los casos para ello se hizo una implementación en *Python* de Runge Kutta de orden 4, como el que se puede encontrar en el Apéndice C.

#### 4.1.2. Código para calcular las curvaturas

Utilizando las funciones para calcular los símbolos de Christoffel se desarrollaron otras 3 funciones para calcular las componentes de los distintos tensores curvatura.

#### Tensor de Riemann

Para calcular las componentes del tensor de Riemann se ha utilizado la fórmula (3.56). Para ahorrar tiempo de cálculo, se calculan previamente las derivadas parciales de los símbolos de



Christoffel, como se puede ver en la función *derivada\_simbolos\_christoffel*, a la que se le pasan los símbolos de Christoffel y las variables que hay en la métrica. Posteriormente ya se calculan las componentes del tensor de Riemann.

---

```

1 def derivada_simbolos_christoffel(v_simbolos_christoffel, variables):
2     matriz_diff_simbolos= [ [ [ [0 for k in range(len(variables))]
3         for j in range(len(variables))] for i in range(len(variables))] for l in range(len(vari
4
5     for v in range(len(variables)):
6         for i in range(len(variables)):
7             for j in range(len(variables)):
8                 for k in range(len(variables)):
9                     #Para comprobar que lo que se va a derivar es una función
10                    if type(v_simbolos_christoffel[i][j][k]) in
11                        [ sp.core.mul.Mul ,sp.core.power.Pow,sp.core.add.Add]:
12                        matriz_diff_simbolos[i][j][k][v] =sp.simplify(
13                            v_simbolos_christoffel[i][j][k].diff(variables[v]))
14    return matriz_diff_simbolos
15 def componentes_tensor_riemann(g, variables):
16     #Calculo los simbolos de Christoffel
17     v_simbolos_christoffel = simbolos_christoffel(g,variables)
18     #Matriz con las derivadas parciales de los simbolos de Christoffel
19     matriz_diff_simbolos = derivada_simbolos_christoffel(v_simbolos_christoffel, variables)
20
21     # n = len(variables)
22     #Matriz n^4
23     R = [ [ [ [0 for k in range(len(variables))]
24         for j in range(len(variables))] for i in range(len(variables))] for l in range(len(vari
25
26     for l in range(len(variables)):
27         for i in range(len(variables)):
28             for j in range(len(variables)):
29                 for k in range(len(variables)):
30
31                     R[l][i][j][k] += matriz_diff_simbolos[l][j][k][i]
32
33                     R[l][i][j][k] +=- matriz_diff_simbolos[l][i][k][j]
34                     #Calculo las sumas
35                     s_1 = 0
36                     s_2 = 0

```

```

37         #Sumatorio positivo
38         for m in range(len(variables)):
39             s_1 += v_simbolos_christoffel[m][j][k]*
40                 v_simbolos_christoffel[l][i][m]
41         #Sumatorio negativo
42         for m in range(len(variables)):
43             s_2 += v_simbolos_christoffel[m][i][k]*
44                 v_simbolos_christoffel[l][j][m]
45
46         R[l][i][j][k] += s_1 -s_2
47         R[l][i][j][k] = sp.simplify(R[l][i][j][k])
48     return R

```

---

## Tensor curvatura

El tensor curvatura se puede calcular de dos formas, utilizando la definición del tensor curvatura, como se puede ver en siguiente código

---

```

1 def componentes_tensor_curvatura(g, variables):
2     tensor_riemann = componentes_tensor_riemann(g, variables)
3
4     tensor_curvatura= [ [ [ [0 for l in range(len(variables))] for k in range(len(variables))
5
6
7     g = np.array(g)
8
9     for i in range(len(variables)):
10        for j in range(len(variables)):
11            for k in range(len(variables)):
12                for l in range(len(variables)):
13
14                    for m in range(len(variables)):
15                        if tensor_riemann[m][i][j][k] != 0 and g[m][l] != 0:
16                            tensor_curvatura[i][j][k][l] += tensor_riemann[m][i][j][k]*g[m]
17
18                tensor_curvatura[i][j][k][l] = sp.simplify(tensor_curvatura[i][j][k][l])
19     return tensor_curvatura

```

---

Un gran problema que tiene el código es el tiempo que tarda en calcular las componentes, porque para ello tiene que previamente haber calculado los símbolos de Christoffel. Una mejora podría ser el uso de la proposición 3.7.6, pero no proporciona una mejora significativa. La mejor solución es el uso de la ecuación (3.65) que es mucho más simple computacionalmente hablando, además que previamente calculamos las segundas derivadas parciales de las componentes de la métrica para ahorrar tiempo.

---

```

1 def diff_g(g, variables):
2     g_diff = [ [ [0 for v in range(len(variables))]
3                 for j in range(len(g))] for i in range(len(g))]
4     for i in range(len(g)):
5         for j in range(len(g)):
6             for v in range(len(variables)):
7                 g_diff[i][j][v] = g[i][j].diff(variables[v])
8
9     g_2_diff = [ [ [ [0 for vv in range(len(variables))]
10                  for v in range(len(variables))] for j in range(len(g))] for i in range(len(g))]
11
12    for i in range(len(g)):
13        for j in range(len(g)):
14            for k in range(len(variables)):
15                for v in range(len(variables)):
16                    g_2_diff[i][j][k][v] = g_diff[i][j][v].diff(variables[v])
17
18    return g_2_diff
19 def componentes_tensor_curvatura(g, variables):
20
21    tensor_curvatura= [ [ [ [0 for l in range(len(variables))]
22                        for k in range(len(variables))] for j in range(len(variables))]
23                    for i in range(len(variables))]
24
25    g = np.array(g)
26
27    g_diff = diff_g(g, variables)
28
29    for i in range(len(variables)):
30        for j in range(len(variables)):
31            for k in range(len(variables)):

```

```

32         for l in range(len(variables)):
33             tensor_curvatura[i][j][k][l]= 1/2*(g_diff[j][l][k][i]
34                 -g_diff[i][l][k][j]
35                 - g_diff[j][k][l][i]
36                 + g_diff[i][k][l][j])
37     return tensor_curvatura

```

---

## Tensor curvatura de Ricci

Para calcular las componentes del tensor de curvatura de Ricci se desarrolló una función siguiendo la ecuación (3.72), como se puede ver en el siguiente código utilizando únicamente las componentes del tensor de Riemann.

```

1  def componentes_tensor_ricci(g,variables):
2      R = componentes_tensor_riemann(g, variables)
3
4      Ricci_curv= [ [0 for i in range(len(variables))] for j in range(len(variables))]
5
6      for i in range(len(variables)):
7          for j in range(len(variables)):
8              for k in range(len(variables)):
9                  Ricci_curv[i][j] += R[k][k][i][j]
10     return Ricci_curv

```

---

## Curvatura escalar

Para calcular la curvatura escalar se escribió una función a partir de la ecuación (3.73) usando las funciones previamente mencionadas del cálculo de la inversa matricial y las componentes del tensor de Ricci.

```

1  def curvatura_escalar(g, variables):
2      g_inv= inversa(g)

```

```

3     R = componentes_tensor_ricci(g,variables)
4     S = 0
5     for i in range(len(variables)):
6         for j in range(len(variables)):
7             S += sp.simplify(g_inv[i][j]*R[i][j])
8     return S

```

---

Todo este código se puede ver y descargar en el repositorio de GitHub <https://github.com/FernanGI/TFG>.

### 4.1.3. Ejemplo de la ejecución de las funciones

#### Ejemplo métrica esfera 2 dimensional

Que mejor forma de introducir el funcionamiento de la librería que con el ejemplo de la esfera 2 dimensional, como hicimos en el ejemplo 3.5.6. Cuya métrica es:

$$g = dr \otimes dr + r^2 d\theta \otimes d\theta + r^2 \sin(\theta)^2 d\phi \otimes d\phi \quad (4.2)$$

La forma matricial de esta métrica es:

$$g = \begin{pmatrix} r^2 & 0 \\ 0 & r^2 \sin^2(\theta) \end{pmatrix}$$

Y la matriz inversa:

$$\begin{pmatrix} \frac{1}{r^2} & 0 \\ 0 & \frac{1}{r^2 \sin^2(\theta)} \end{pmatrix} \quad (4.3)$$

Entonces el cálculo de los símbolos de Christoffel con la función se puede ver en la figura 4.1. Una vez ya se han calculado los símbolos de Christoffel podemos calcular las componentes del tensor curvatura de Riemann, como se puede ver en la imagen 4.2. Luego se puede calcular el tensor curvatura de la esfera, figura 4.3. En ella se puede observar el resultado de la proposición 3.7.6. Y finalmente las componentes del tensor curvatura de Ricci y la curvatura escalar, como se puede ver en la figura 4.4. Un resultado curioso que se obtiene de la curvatura escalar es que si recordamos la curva de Gauss de una esfera es  $\frac{1}{r^2}$ , el cual es muy similar a la escalar. Esto es porque la curvatura escalar es el doble que la curvatura de Gauss. [12]

### Símbolos de Chistoffel

```
In [18]: 1 v simbolos_christoffel=simbolos_christoffel(g,variables)
2 for i in range(len(v_simbolos_christoffel)):
3     for j in range(len(v_simbolos_christoffel)):
4         for k in range(len(v_simbolos_christoffel)):
5             #if v_simbolos_christoffel[i][j][k] != 0:
6                 display(Math(f'\Gamma^{{{variables[i]}}}_{{{variables[j]},{variables[k]}}}' + latex(v_simbolos_christoffel[i][j][k])))

\Gamma^{\theta}_{\theta,\theta} = 0
\Gamma^{\theta}_{\theta,\phi} = 0
\Gamma^{\theta}_{\phi,\theta} = 0
\Gamma^{\theta}_{\phi,\phi} = -0.5 \sin(2\theta)
\Gamma^{\phi}_{\theta,\theta} = 0
\Gamma^{\phi}_{\theta,\phi} = \frac{1.0}{\tan(\theta)}
\Gamma^{\phi}_{\phi,\theta} = \frac{1.0}{\tan(\theta)}
\Gamma^{\phi}_{\phi,\phi} = 0
```

Figura 4.1: Símbolos de Christoffel de la esfera.

### Componentes del tensor de Riemann

```
In [19]: 1 R = componentes_tensor_riemann(g, variables)
2 for l in range(len(variables)):
3     for i in range(len(variables)):
4         for j in range(len(variables)):
5             for k in range(len(variables)):
6                 if R[l][i][j][k] != 0:
7                     display(Math(f'R^{{{variables[l]}}}_{{{variables[i]},{variables[j]},{variables[k]}}}' + latex(R[l][i][j][k])))

R^{\theta}_{\theta,\theta,\phi} = \frac{0.5 \sin(2\theta)}{\tan(\theta)} - 1.0 \cos(2\theta)
R^{\theta}_{\phi,\theta,\phi} = -\frac{0.5 \sin(2\theta)}{\tan(\theta)} + 1.0 \cos(2\theta)
R^{\phi}_{\theta,\phi,\theta} = -1.0
R^{\phi}_{\phi,\theta,\theta} = 1.0
```

Figura 4.2: Componentes del tensor de Riemann de la esfera

## Ejemplo métrica Warp Drive

Para ilustrar el funcionamiento del cálculo de las geodésicas utilizaremos la métrica del Warp Drive de Miguel Alcubierre, también conocida como métrica de Alcubierre: [1]

$$ds^2 = -c^2 dt^2 + (dx - v_s f(r_s) dt)^2 + dy^2 + dz^2 \quad (4.4)$$

Donde:

$$\begin{aligned} v_s &= \frac{dx_s(t)}{dt} \\ r_s(t) &= \sqrt{(x - x_s(t))^2 + y^2 + z^2} \\ f(r_s) &= \frac{\tanh(\sigma(r_s + R)) - \tanh(\sigma(r_s - R))}{2 \tanh(\sigma R)} \end{aligned} \quad (4.5)$$

### Componentes tensor curvatura

```
In [20]: 1 R = componentes_tensor_curvatura(g,variables)
2 for l in range(len(variables)):
3     for i in range(len(variables)):
4         for j in range(len(variables)):
5             for k in range(len(variables)):
6                 if R[l][i][j][k] != 0:
7                     display(Math('R_{'+variables[l]+''+variables[i]+''+variables[j]+''+variables[k]+''} = '+l
```

$$R_{\theta\theta,\phi\theta,\phi\theta} = -1.0r^2 \sin^2(\theta)$$

$$R_{\theta\theta,\phi\phi,\theta\theta} = r^2 \left( \frac{0.5 \sin(2\theta)}{\tan(\theta)} - 1.0 \cos(2\theta) \right)$$

$$R_{\phi\theta,\theta\theta,\phi\theta} = 1.0r^2 \sin^2(\theta)$$

$$R_{\phi\theta,\theta\theta,\theta\theta} = r^2 \left( -\frac{0.5 \sin(2\theta)}{\tan(\theta)} + 1.0 \cos(2\theta) \right)$$

Figura 4.3: Componentes del tensor curvatura de la esfera

### Componentes tensor curvatura Ricci

```
In [18]: 1 Ricci_curv = componentes_tensor_ricci(g,variables)
2 for i in range(len(variables)):
3     for j in range(len(variables)):
4         if Ricci_curv[i][j] != 0:
5             display(Math('R_{'+variables[i]+''+variables[j]+''} = '+latex(Ricci_curv[i][j]))
```

$$R_{\theta\theta,\theta\theta} = 1.0$$

$$R_{\phi\theta,\phi\theta} = \frac{0.5 \sin(2\theta)}{\tan(\theta)} - 1.0 \cos(2\theta)$$

### Curvatura escalar

```
In [22]: 1 curvatura_escalar(g,variables)
```

Out[22]:  $\frac{2.0}{r^2}$

Figura 4.4: Cálculo del tensor de curvatura de Ricci y la curvatura escalar

Donde los parámetros  $R > 0$  es el radio de la *warp bubble* y  $\sigma > 0$  su grosor, pero para mantener una notación simple se seguirá utilizando  $v_s$  y  $f(r_s)$ . Esta métrica representa un objeto, la *warp bubble*, moviéndose por un espacio-tiempo y este objeto teóricamente puede viajar más rápido que la velocidad de la luz, pero el objetivo de este apartado no es discutir sobre el significado físico de esta métrica sino el análisis de los símbolos de Christoffel con la librería desarrollada. Entonces su expresión en forma matricial es:

$$\begin{bmatrix} -c^2 + v_s^2 f^2(x, y, z, t) & -v_s f(x, y, z, t) & 0 & 0 \\ -v_s f(x, y, z, t) & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

Y su matriz inversa es:

$$\begin{bmatrix} -\frac{1}{c^2} & -\frac{v_s f(x,y,z,t)}{c^2} & 0 & 0 \\ -\frac{v_s f(x,y,z,t)}{c^2} & 1 - \frac{v_s^2 f^2(x,y,z,t)}{c^2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.7)$$

Entonces introducimos forma matricial en Python, previamente habiendo mencionado las variables y las constantes que hay en la métrica: Entonces calculamos los símbolos de Christoffel:

```
In [5]: 1 #Declaración de las variables que vamos a usar
2 t, x, y, z = sp.symbols('t x y z')
3 #Esto aunque son variables vamos a ponerlas como variables para que sea mas elegante
4 c = sp.symbols("c")
5 #Vector con las variables que vamos a usar
6 variables = [t,x,y,z]

In [6]: 1 #Funciones que aparecen en la métrica
2 f = sp.Function('f')(x,y,z,t)
3 v_s = sp.symbols("v_s")

In [7]: 1 #Expreso la métrica en forma matricial
2 g = sp.Matrix([[ -c**2+f**2*v_s**2,-v_s*f,0,0],
3               [-v_s*f,1,0,0],
4               [0,0,1,0],
5               [0,0,0,1]])
6 g

Out[7]: 
$$\begin{bmatrix} -c^2 + v_s^2 f^2(x, y, z, t) & -v_s f(x, y, z, t) & 0 & 0 \\ -v_s f(x, y, z, t) & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```

Figura 4.5: Definición de la matriz de la métrica de Alcubierre

Entonces los símbolos de Christoffel son:

$$\begin{aligned} \Gamma_{tt}^t &= \frac{f^2 f_x v^3}{c^2}, \Gamma_{t,t}^z = -f f_y v^2, \quad \Gamma_{tt}^y = -f f_z v^2, \quad \Gamma_{tt}^x = \frac{f^3 f_x v^4 - c^2 f f_x v_s^2 - c^2 f_t v_s}{c^2}, \\ \Gamma_{tx}^t &= -\frac{f f_x v_s^2}{c^2}, \Gamma_{tx}^x = -\frac{f^2 f_x v_s^3}{c^2}, \quad \Gamma_{tx}^y = \frac{f_y v_s}{2}, \quad \Gamma_{tx}^z = \frac{f_z v}{2} \\ \Gamma_{ty}^t &= -\frac{f f_y v^2}{2c^2}, \quad \Gamma_{ty}^x = -\frac{f^2 f_y v_s^3 + c^2 f_y v_s}{2c^2}, \quad \Gamma_{ty}^z = -\frac{f f_z v_s^2}{2c^2}, \quad \Gamma_{tz}^x = -\frac{f^2 f_z v_s^3 + c^2 f_z v_s}{2c^2} \\ \Gamma_{xx}^t &= \frac{f_x v_s}{c^2}, \quad \Gamma_{xx}^x = \frac{f f_x v_s^2}{c^2}, \quad \Gamma_{xy}^t = \frac{f_y v_s}{2c^2}, \quad \Gamma_{xy}^x = \frac{f f_y v_s^2}{2c^2}, \\ \Gamma_{xz}^t &= \frac{f_z v_s}{2c^2}, \quad \Gamma_{xz}^x = \frac{f f_z v_s^2}{2c^2} \end{aligned} \quad (4.8)$$



```
In [8]: 1 v_simbolos_christoffel=simbolos_christoffel(g,variables)
2 for i in range(len(v_simbolos_christoffel)):
3     for j in range(len(v_simbolos_christoffel)):
4         for k in range(len(v_simbolos_christoffel)):
5             if v_simbolos_christoffel[i][j][k] != 0:
6                 display(Math(r'\Gamma^{{{variables[i]}}_{{{variables[j]},{variables[k]}}}') = ' + latex(v_simbolos_ch
```

$$\Gamma_{t,t}^x = \frac{v_3^3 f^2(x, y, z, t) \frac{\partial}{\partial x} f(x, y, z, t)}{c^2}$$

$$\Gamma_{t,x}^x = -\frac{1.0 v_3^2 f(x, y, z, t) \frac{\partial}{\partial x} f(x, y, z, t)}{c^2}$$

$$\Gamma_{t,y}^x = -\frac{0.5 v_3^2 f(x, y, z, t) \frac{\partial}{\partial y} f(x, y, z, t)}{c^2}$$

$$\Gamma_{t,z}^x = -\frac{0.5 v_3^2 f(x, y, z, t) \frac{\partial}{\partial z} f(x, y, z, t)}{c^2}$$

$$\Gamma_{x,t}^x = -\frac{1.0 v_3^2 f(x, y, z, t) \frac{\partial}{\partial x} f(x, y, z, t)}{c^2}$$

$$\Gamma_{x,x}^x = \frac{1.0 v_3 \frac{\partial}{\partial x} f(x, y, z, t)}{c^2}$$

$$\Gamma_{x,y}^x = \frac{0.5 v_3 \frac{\partial}{\partial y} f(x, y, z, t)}{c^2}$$

$$\Gamma_{x,z}^x = \frac{0.5 v_3 \frac{\partial}{\partial z} f(x, y, z, t)}{c^2}$$

$$\Gamma_{y,t}^x = -\frac{0.5 v_3^2 f(x, y, z, t) \frac{\partial}{\partial y} f(x, y, z, t)}{c^2}$$

$$\Gamma_{y,x}^x = \frac{0.5 v_3 \frac{\partial}{\partial y} f(x, y, z, t)}{c^2}$$

$$\Gamma_{z,t}^x = -\frac{0.5 v_3^2 f(x, y, z, t) \frac{\partial}{\partial z} f(x, y, z, t)}{c^2}$$

$$\Gamma_{z,x}^x = \frac{0.5 v_3 \frac{\partial}{\partial z} f(x, y, z, t)}{c^2}$$

$$\Gamma_{t,t}^y = -1.0 v_3^2 f(x, y, z, t) \frac{\partial}{\partial x} f(x, y, z, t) - 1.0 v_3 \frac{\partial}{\partial t} f(x, y, z, t) + \frac{1.0 v_3^4 f^3(x, y, z, t) \frac{\partial}{\partial x} f(x, y, z, t)}{c^2}$$

$$\Gamma_{t,x}^y = -\frac{1.0 v_3^3 f^2(x, y, z, t) \frac{\partial}{\partial x} f(x, y, z, t)}{c^2}$$

$$\Gamma_{t,y}^y = -\frac{0.5 v_3 (c^2 + v_3^2 f^2(x, y, z, t)) \frac{\partial}{\partial y} f(x, y, z, t)}{c^2}$$

$$\Gamma_{t,z}^y = -\frac{0.5 v_3 (c^2 + v_3^2 f^2(x, y, z, t)) \frac{\partial}{\partial z} f(x, y, z, t)}{c^2}$$

$$\Gamma_{x,t}^y = -\frac{1.0 v_3^3 f^2(x, y, z, t) \frac{\partial}{\partial x} f(x, y, z, t)}{c^2}$$

$$\Gamma_{x,x}^y = \frac{1.0 v_3^2 f(x, y, z, t) \frac{\partial}{\partial x} f(x, y, z, t)}{c^2}$$

$$\Gamma_{x,y}^y = \frac{0.5 v_3^2 f(x, y, z, t) \frac{\partial}{\partial y} f(x, y, z, t)}{c^2}$$

$$\Gamma_{x,z}^y = \frac{0.5 v_3^2 f(x, y, z, t) \frac{\partial}{\partial z} f(x, y, z, t)}{c^2}$$

$$\Gamma_{y,t}^y = -\frac{0.5 v_3 (c^2 + v_3^2 f^2(x, y, z, t)) \frac{\partial}{\partial y} f(x, y, z, t)}{c^2}$$

$$\Gamma_{y,x}^y = \frac{0.5 v_3^2 f(x, y, z, t) \frac{\partial}{\partial y} f(x, y, z, t)}{c^2}$$

Figura 4.6: Símbolos de Christoffel de la métrica de Alcubierre

$$\begin{aligned}
\Gamma_{z,t}^x &= -\frac{0.5v_s (c^2 + v_s^2 f^2(x, y, z, t)) \frac{\partial}{\partial z} f(x, y, z, t)}{c^2} \\
\Gamma_{z,x}^x &= \frac{0.5v_s^2 f(x, y, z, t) \frac{\partial}{\partial z} f(x, y, z, t)}{c^2} \\
\Gamma_{t,t}^y &= -1.0v_s^2 f(x, y, z, t) \frac{\partial}{\partial y} f(x, y, z, t) \\
\Gamma_{t,x}^y &= 0.5v_s \frac{\partial}{\partial y} f(x, y, z, t) \\
\Gamma_{x,t}^y &= 0.5v_s \frac{\partial}{\partial y} f(x, y, z, t) \\
\Gamma_{t,t}^z &= -1.0v_s^2 f(x, y, z, t) \frac{\partial}{\partial z} f(x, y, z, t) \\
\Gamma_{t,x}^z &= 0.5v_s \frac{\partial}{\partial z} f(x, y, z, t) \\
\Gamma_{x,t}^z &= 0.5v_s \frac{\partial}{\partial z} f(x, y, z, t)
\end{aligned}$$


---

Figura 4.7: Símbolos de Christoffel de la métrica de Alcubierre

## 4.2. Estudio de las curvas geodésicas y la curvatura en diferentes métricas

Entonces se ha utilizado este programa con alguna las diferentes métricas que se han ido mencionando durante este trabajo, como puede ser la métrica de Poincaré, y finalmente se estudian las métricas de Minkowski y de Schwarzschild.

### 4.2.1. Métrica de Poincaré

Como se mostró en el ejemplo 3.5.5 la métrica de Poincaré está definida de la siguiente forma:

$$g = \frac{dx \otimes dx + dy \otimes dy}{y^2} \quad (4.9)$$

Sobre la variedad diferencial  $\mathbb{R} \times \mathbb{R}^+$ .

Su expresión matricial es:

$$\begin{bmatrix} \frac{1}{y^2} & 0 \\ 0 & \frac{1}{y^2} \end{bmatrix} \quad (4.10)$$

Ahora vamos a ver que la métrica de Poincaré es una métrica pseudo-riemanniana observando

la expresión matricial, en ella podemos observar que es no degenerada debido a que la variedad sobre la que está definida no tiene puntos que hagan que  $\frac{1}{y^2}$  sea singular por lo tanto será no singular. Y además será simétrica ya que los términos cruzados son iguales.

Una vez se ha visto que es una variedad pseudo-riemanniana vamos a introducir la métrica en el programa, que hemos realizado para calcular primero las derivadas covariantes y posteriormente los símbolos de Christoffel, como se puede observar en la figura 4.8.

Entonces con el programa que se ha mostrado antes obtenemos que los símbolos de Christoffel son:

$$\begin{aligned}\Gamma_{x,x}^x &= \Gamma_{y,y}^x = \Gamma_{x,y}^y = \Gamma_{y,x}^y = 0 \\ \Gamma_{x,y}^x &= \Gamma_{y,x}^x = \Gamma_{y,y}^y = \frac{-1}{y} \\ \Gamma_{x,x}^y &= \frac{1}{y}\end{aligned}\tag{4.11}$$

Para calcular las ecuaciones de las geodésicas también vamos a utilizar el programa que hemos mencionado previamente, como se puede observar en la figura 4.9.

Entonces, obtenemos que las geodésicas tienen la siguiente forma:

$$\begin{aligned}\frac{d^2x}{d\lambda^2} - \frac{2}{y} \frac{dx}{d\lambda} \frac{dy}{d\lambda} &= 0 \\ \frac{d^2y}{d\lambda^2} + \frac{1}{y} \left( \frac{dx}{d\lambda}^2 - \frac{dy}{d\lambda}^2 \right) &= 0\end{aligned}\tag{4.12}$$

Se obtienen las siguientes clases de geodésicas:

- Si tomamos un punto de la forma  $(0,y)$  con  $y \in \mathbb{R}$  y una velocidad nula en el eje  $x$ , es decir,  $x = \dot{x} = 0$  entonces obtenemos el sistema de ecuaciones:

$$\begin{aligned}\frac{d^2x}{d\lambda^2} &= 0 \\ \frac{d^2y}{d\lambda^2} - \frac{1}{y} \frac{dy}{d\lambda}^2 &= 0\end{aligned}\tag{4.13}$$

Cuyas soluciones son  $(x(t), y(t)) = (0, y_0 e^{\frac{y_0}{2} t})$  que son líneas rectas sobre el eje  $y$  y como la que se puede observar en la figura 4.10.

- En cualquier otro caso si tomamos un punto arbitrario,  $(x_0, y_0)$ , y un vector,  $(\dot{x}_0, \dot{y}_0) \in T_{(x_0, y_0)}\mathcal{M}$ . Entonces, integramos la primera ecuación de las geodésicas de la métrica de

```

In [3]: 1 #Declaración de las variables que vamos a usar
2 x, y = sp.symbols('x y')
3 #Vector con las variables que vamos a usar
4 variables = [x,y]

In [4]: 1 #Expreso la métrica en forma matricial
2 g = sp.Matrix([[1/y**2,0],[0,1/y**2]])
3 g

Out[4]: 
$$\begin{bmatrix} \frac{1}{y^2} & 0 \\ 0 & \frac{1}{y^2} \end{bmatrix}$$


In [5]: 1 #Esto es para poder hacerlo en python y no en sympy que es horrible
2 g = [[1/y**2,0],[0,1/y**2]]
3
4 #SOLO SIRVE PARA MATRICES DIAGONALES
5 g_inv = [[i for i in v] for v in g]
6 for i in range(len(g_inv)):
7     for j in range(len(g_inv)):
8         if g_inv[i][j] != 0:
9             g_inv[i][i] = g_inv[i][i]**-1
10 #Para ver bonita la matriz inversa
11 g_inv_sp = Matrix(g_inv)
12 g_inv_sp
13

Out[5]: 
$$\begin{bmatrix} y^2 & 0 \\ 0 & y^2 \end{bmatrix}$$


In [6]: 1 #Este vector va a contener la matriz g derivada respecto de cada la variable
2 #Por ejemplo derivadas_variable[0] es la derivada de toda la matriz respecto de t
3 # derivadas_variable[1] es la derivada de toda la matriz respecto de x
4 #Y derivadas_variable[i][0][0] es la derivad parcial de g_{0,0} respecto x
5 # y así sucesivamente
6 derivadas_variable = []
7
8 for variable in variables:
9     derivada_variable = [[0 for i in variables] for j in variables]
10    for i in range(len(variables)):
11        for j in range(len(variables)):
12            funcion = g[i][j]
13            #Para no hacer la derivada de un número
14            if type(funcion) == sp.core.mul.Mul or type(funcion) == sp.core.power.Pow:
15                derivada_variable[i][j] = simplify(funcion.diff(variable))
16
17    derivadas_variable.append(derivada_variable)
18 derivadas_variable

Out[6]: 
$$\left[ \left[ [0, 0], [0, 0] \right], \left[ \left[ -\frac{2}{y^3}, 0 \right], \left[ 0, -\frac{2}{y^3} \right] \right] \right]$$


In [7]: 1
2
3 #Esta matriz va a tener los simbolos de chistoffle
4 #Por ejemplo, simbolo_chirtoffle[2][0][1] seria \gamma^y_{t,x}
5 simbolos_chistoffle = [[[0 for j in variables] for i in variables] for m in variables]
6
7
8 for m in range(len(variables)):
9     for i in range(len(variables)):
10        for j in range(len(variables)):
11            simbolo = 0
12            #El sumatorio respecto del indice libre l
13            for l in range(len(variables)):
14                #Para evitar hacer una multiplicación por 0
15                if g_inv[m][l] != 0:
16                    simbolo = simbolo + g_inv[m][l]*(derivadas_variable[j][i][l]+derivadas_variable[i][j][l]-der
17            #Aquí multiplico por el 1/2 que hemos sacado de factor común
18            simbolo = 1/2*simbolo
19            simbolos_chistoffle[m][i][j] = simplify(simbolo)

```

Figura 4.8: Calculo de los símbolos de Christoffel

Poincaré:

$$\begin{aligned}
 \frac{d^2 x}{d\lambda^2} - \frac{2}{y} \frac{dx}{d\lambda} \frac{dy}{d\lambda} &= 0 \\
 \frac{1}{\frac{dx}{d\lambda}} \frac{d^2 x}{d\lambda^2} &= \frac{2}{y} \frac{dy}{d\lambda} \\
 \int_{x_0}^{\dot{x}} \frac{1}{\frac{dx}{d\lambda}} \frac{d^2 x}{d\lambda^2} &= \int_{y_0}^y \frac{2}{y} \frac{dy}{d\lambda} \\
 \log\left(\frac{dx}{d\lambda}\right) \Big|_{x_0}^{\dot{x}} &= \log(y^2) \Big|_{y_0}^y \\
 \log\left(\frac{\dot{x}}{x_0}\right) &= \log\left(\frac{y^2}{y_0^2}\right) + c \\
 \dot{x} &= \frac{y^2 C \dot{x}_0}{y_0}
 \end{aligned}
 \tag{4.14}$$

```

In [11]: 1 #Variable para poder derivar
          2 tau = sp.symbols("tau")
          3
          4 #Defino las derivadas, esto para poder resolverlo
          5 dx = sp.Function("dx")(tau)
          6 dy = sp.Function("dy")(tau)
          7
          8 #Quitamos dz
          9 variables_derivadas = [dx,dy]
          10

In [12]: 1 geodesicas = [i.diff(tau) for i in variables_derivadas]
          2
          3 for variable in range(len(variables)):
          4     for i in range(len(variables)):
          5         for j in range(len(variables)):
          6             geodesicas[variable] = geodesicas[variable] + simplify( simbolos_chistoffle[variable][i][j]*variable
          7             geodesicas[variable] = sp.Eq(geodesicas[variable],0)
          8
          9

In [13]: 1 for v in geodesicas:
          2     display(v)
          3
          4 
$$\frac{d}{d\tau}dx(\tau) - \frac{2.0 dx(\tau) dy(\tau)}{y} = 0$$

          5
          6 
$$\frac{d}{d\tau}dy(\tau) + \frac{1.0 dx^2(\tau)}{y} - \frac{1.0 dy^2(\tau)}{y} = 0$$


```

Figura 4.9: Cálculo de las geodésicas con el programa

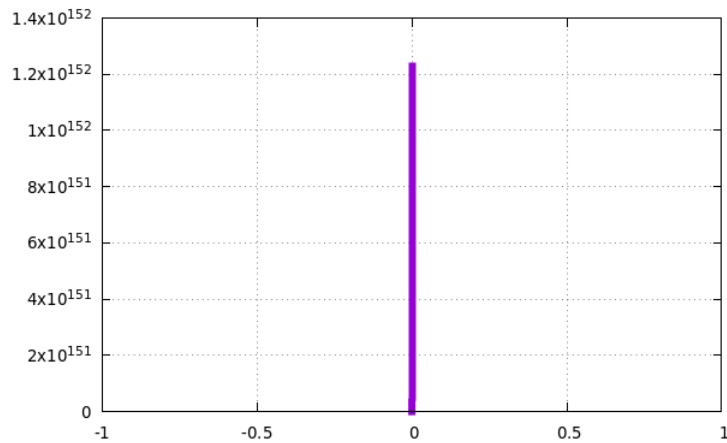


Figura 4.10: Geodésica en la métrica de Poincaré con el punto inicial (0,1) y velocidades (0, 10)

Entonces, por la proposición 3.6.11 obtenemos que:

$$\frac{dx^2}{d\lambda} + \frac{dy^2}{d\lambda} = K \tag{4.15}$$

Utilizando el resultado de la ecuación (4.14):

$$\begin{aligned}\frac{dx^2}{d\lambda} + \frac{dy^2}{d\lambda} &= y^2 K \\ \frac{dy^2}{d\lambda} &= y^2 K - \left(\frac{y^2 C \dot{x}_0}{y_0}\right)^2 \\ \frac{dy}{d\lambda} &= \sqrt{y^2 K - \left(\frac{y^2 C \dot{x}_0}{y_0}\right)^2}\end{aligned}\tag{4.16}$$

Las curvas geodésicas vienen determinadas por este sistema de ecuaciones diferenciales:

$$\begin{aligned}\frac{dx}{d\lambda} &= y^2 C \\ \frac{dy}{d\lambda} &= y \sqrt{K - y^2 \left(\frac{C \dot{x}_0}{y_0}\right)^2}\end{aligned}\tag{4.17}$$

Se obtienen semicírculos, como se puede ver en la figura 4.11.

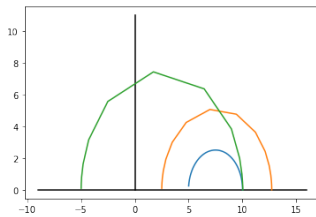


Figura 4.11: Diferentes geodésicas semicirculares en la métrica de Poincaré

Ahora vamos a estudiar la curvatura de esta variedad. Para ello vamos a ver primero las componentes del tensor curvatura de Riemann:

$$\begin{aligned}R_{x,y,y}^x &= -\frac{1,0}{y^2} \\ R_{y,x,y}^x &= \frac{1,0}{y^2} \\ R_{x,y,x}^y &= \frac{1,0}{y^2} \\ R_{y,x,x}^y &= -\frac{1,0}{y^2}\end{aligned}\tag{4.18}$$

Y las componentes del tensor curvatura son:

$$\begin{aligned}
 R_{x,y,x,y} &= \frac{1,0}{y^4} \\
 R_{x,y,y,x} &= -\frac{1,0}{y^4} \\
 R_{y,x,x,y} &= -\frac{1,0}{y^4} \\
 R_{y,x,y,x} &= \frac{1,0}{y^4}
 \end{aligned}
 \tag{4.19}$$

En ellas se puede observar las propiedades de simetría y antisimétrica de este tensor. Además de la redundancia de información que hay, ya que son iguales y lo único que cambia es el signo. Las componentes del tensor de Ricci serán:

$$\begin{aligned}
 R_{x,x} &= -\frac{1,0}{y^2} \\
 R_{y,y} &= -\frac{1,0}{y^2}
 \end{aligned}
 \tag{4.20}$$

Se puede observar que lo único que importa a la hora de determinar las componentes del tensor curvatura de este espacio es la cercanía de la coordenada  $y$  a la origen. Para ello vamos a ver, por seleccionar una de las 3, como se comporta el tensor curvatura cuando se aleja o se acerca la coordenada y del origen.

$$\begin{aligned}
 \lim_{y \rightarrow 0^+} \frac{1}{y^4} &= \frac{1}{0} = \infty \\
 \lim_{y \rightarrow \infty} \frac{1}{y^4} &= \frac{1}{\infty} = 0
 \end{aligned}
 \tag{4.21}$$

Lo que nos viene a decir es que cuanto más cerca esté la coordenada  $y$  del origen, más distorsionado esté el tensor curvatura, y cuanto más alejada esté, se asemejará más a un espacio plano.

Su curvatura escalar en este caso es de -2, es decir, como su propio nombre indica esta métrica determina un espacio hiperbólico.

#### 4.2.2. Métrica de Minkowski

En 1908 Hermann Minkowski (1864-1909), un matemático alemán, presentó la idea de juntar el espacio y el tiempo en un mismo objeto y lo llamó "espaciotiempo".

**Definición 4.2.1** (Ver [3]). Se llama **espacio-tiempo** a una variedad semi-riemanniana de cuatro dimensiones cuya métrica en cada punto tiene índice 1.

En este espacio-tiempo los conceptos matemáticos que hemos visto a lo largo del trabajo se transforman en conceptos físicos: las geodésicas nulas se convierten en fotones, las geodésicas temporales se convierten en partículas materiales, al conjunto de geodésicas nulas y temporales se les llaman líneas del universo, o a la métrica se le llaman gravedad ...

A su vez aparece un nuevo concepto el **tiempo propio** que es un nuevo parámetro, que se denota por  $\tau$ , de la geodésica temporal parametrizada de tal forma que sea una geodésica futura, temporal y su vector tangente,  $\vec{v} = \vec{v}(\tau)$  que verifique que  $g(\vec{v}(\tau), \vec{v}(\tau)) = -1$ . Lo que viene a representar este tiempo propio es el tiempo que medido por un observador que viaja con la partícula.

Esta métrica representa un espacio tiempo vacío, sin ningún objeto que perturbe su forma. La métrica de Minkowski es:

$$g = -c^2 dt \otimes dt + dx \otimes dx + dy \otimes dy + dz \otimes dz \quad (4.22)$$

Claramente es una métrica semi-Riemanniana, si nos vamos a la definición 3.5.2 podemos observar que:

- Es simétrica ya que si nos vamos a su expresión matricial de la métrica es una matriz diagonal.
- Es no degenerada, ya que si observamos la expresión matricial de la métrica se puede ver que claramente es no singular.

Por lo tanto,  $(\mathbb{R}^4, g)$  es una variedad semi-Riemanniana.

Si realizamos el cambio de cartas a coordenadas esféricas tal que  $\mathcal{M} = \mathbb{R} \times \mathbb{R}^+ \times ]0, \pi[ \times ]0, 2\pi[$  que está embebido dentro de  $\mathbb{R}^4$  y las funciones coordenadas:

$$\begin{aligned} t &= t \\ x &= r \cos \theta \sin \varphi \\ y &= r \sin \theta \sin \varphi \\ z &= r \cos \varphi \end{aligned} \quad (4.23)$$

Entonces obtenemos que:

$$\begin{aligned} dt &= dt \\ dx &= (\cos(\theta)\cos(\phi))dr + (-r\sin(\theta)\sin(\phi))d\theta + (-r\cos(\theta)\cos(\phi))d\phi \\ dy &= (\cos(\theta)\sin(\phi))dr + (r\cos(\theta)\cos(\phi))d\theta + (-r\sin(\theta)\cos(\phi))d\phi \\ dz &= \sin(\theta)dr + r\cos(\theta)d\theta \end{aligned} \quad (4.24)$$



Substituyendo el resultado de la ecuación (4.24) en la ecuación (4.22) obtenemos la métrica de Minkowski en coordenadas esféricas:

$$g = -c^2 dt \otimes dt + dr \otimes dr + r^2(d\theta \otimes d\theta + \sin(2\theta)d\varphi \otimes d\varphi) \quad (4.25)$$

Entonces calculamos las geodésicas con el programa mencionado en la sección anterior, como se puede ver en la figura 4.13.

```

In [1]: 1 import sympy as sp
2 from sympy import *
3 import numpy as np
4 #Para resolver sistemas de ecuaciones diferenciales
5 from sympy.solvers.ode.systems import dsolve_system
6 from IPython.display import display, Math

In [2]: 1 sp.interactive.printing.init_printing(use_latex=True)

In [3]: 1 #Declaración de las variables que vamos a usar
2 t, r, theta, phi = sp.symbols('t r theta phi')
3 c = sp.symbols('c')
4
5 #Vector con las variables que vamos a usar
6 variables = [t, r, theta, phi]

In [4]: 1 #Expreso la métrica en forma matricial
2 g = [[-c**2,0,0,0],
3       [0,1,0,0],
4       [0,0,r**2,0],
5       [0,0,0,r**2*sp.sin(theta)**2]]
6 sp.Matrix(g)
7

Out[4]: 
$$\begin{bmatrix} -c^2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & r^2 & 0 \\ 0 & 0 & 0 & r^2 \sin^2(\theta) \end{bmatrix}$$


In [5]: 1 g_inv = [[ g[i][j]**-1 if g[i][j] != 0 else g[i][j] for j in range(len(g))] for i in range(len(g))]
2 sp.Matrix(g_inv)

Out[5]: 
$$\begin{bmatrix} -\frac{1}{c^2} & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 \\ 0 & 0 & \frac{1}{r^2} & 0 \\ 0 & 0 & 0 & \frac{1}{r^2 \sin^2(\theta)} \end{bmatrix}$$


In [60]: 1 simbolos_chistoffle = [[[0 for j in variables] for i in variables] for m in variables]
2
3
4 for m in range(len(variables)):
5     for i in range(len(variables)):
6         for j in range(len(variables)):
7             simbolo = 0
8             #El sumatorio respecto del indice libre l
9             for l in range(len(variables)):
10                #Para evitar hacer una multiplicación por 0
11                if g_inv[m][l] != 0:
12                    simbolo = simbolo + g_inv[m][l]*(derivadas_variable[j][i][l]+derivadas_variable[i][j][l]-der
13                    simbolo = 1/2*simbolo
14                    simbolos_chistoffle[m][i][j] = simplify(simbolo)
15 for i in range(len(simbolos_chistoffle)):
16     for j in range(len(simbolos_chistoffle)):
17         for k in range(len(simbolos_chistoffle)):
18             if simbolos_chistoffle[i][j][k] != 0:
19
20                 display(Math(f'\Gamma^{{{variables[i]}}}_{{{variables[j]}}}_{{{variables[k]}}} = ' + latex(simbolos_

```

$$\Gamma_{\theta,\theta}^r = -1.0r$$

$$\Gamma_{\phi,\phi}^r = -1.0r \sin^2(\theta)$$

$$\Gamma_{r,\theta}^{\theta} = \frac{1.0}{r}$$

$$\Gamma_{\theta,r}^{\theta} = \frac{1.0}{r}$$

$$\Gamma_{\phi,\phi}^{\theta} = -0.5 \sin(2\theta)$$

$$\Gamma_{r,\phi}^{\phi} = \frac{1.0}{r}$$

$$\Gamma_{\theta,\phi}^{\phi} = \frac{1.0}{\tan(\theta)}$$

$$\Gamma_{\phi,r}^{\phi} = \frac{1.0}{r}$$

$$\Gamma_{\phi,\theta}^{\phi} = \frac{1.0}{\tan(\theta)}$$

Figura 4.12: Ejemplo programa cálculo los símbolos de Christoffel

```

In [25]: 1 #Variable para poder derivar
2 tau = sp.symbols("tau")
3
4 #Defino las derivadas, esto para poder resolverlo
5 dt = sp.Function('dt')(tau)
6 dr = sp.Function("dr")(tau)
7 dtheta = sp.Function('dtheta')(tau)
8 dphi = sp.Function('dphi')(tau)
9
10 variables_derivadas = [dt,dr,dtheta,dphi]
11

In [26]: 1 #Esto va a con
2 geodesicas = [i.diff(tau) for i in variables_derivadas]
3
4 for variable in range(len(variables)):
5     for i in range(len(variables)):
6         for j in range(len(variables)):
7             geodesicas[variable] = geodesicas[variable] + simplify( simbolos_chistoffe[i][j]*variable
8             geodesicas[variable] = sp.Eq(geodesicas[variable],0)
9
10

In [46]: 1 for v in geodesicas:
2     display(v)
3

```

$$\begin{aligned}
 \frac{d}{d\tau} dt(\tau) &= 0 \\
 -1.0r \, d\phi^2(\tau) \sin(2\theta) - 1.0r \, d\theta^2(\tau) + \frac{d}{d\tau} dr(\tau) &= 0 \\
 -1.0 \, d\phi^2(\tau) \cos(2\theta) + \frac{d}{d\tau} d\theta(\tau) + \frac{2.0 \, dr(\tau) \, d\theta(\tau)}{r} &= 0 \\
 \frac{2.0 \, d\phi(\tau) \, d\theta(\tau)}{\tan(2\theta)} + \frac{d}{d\tau} d\phi(\tau) + \frac{2.0 \, d\phi(\tau) \, dr(\tau)}{r} &= 0
 \end{aligned}$$

Figura 4.13: Cálculo de las geodésicas

Los símbolos de Christoffel son:

$$\begin{aligned}
 \Gamma_{\theta\theta}^r &= -r, & \Gamma_{\varphi\varphi}^r &= -r \sin^2 \theta, & \Gamma_{r\theta}^\theta &= \Gamma_{\theta r}^\theta = \frac{1}{r} \\
 \Gamma_{\varphi\varphi}^\theta &= -\frac{\sin(2\theta)}{2}, & \Gamma_{r\varphi}^\varphi &= \Gamma_{\varphi r}^\varphi = \frac{1}{r}, & \Gamma_{\varphi\theta}^\varphi &= \Gamma_{\theta\varphi}^\varphi = \cot \theta
 \end{aligned} \tag{4.26}$$

Mientras que si calculamos los símbolos de Christoffel respecto de las coordenadas cartesianas obtendremos que:

$$\Gamma_{v,w}^u = 0 \quad \forall u, v, w \in \{t, x, y, z\} \tag{4.27}$$

Lo cual nos puede chocar un poco ya que los símbolos de Christoffel están relacionados con la curvatura de la variedad, como previamente hemos visto. La expresión de las geodésicas en coordenadas esféricas es:

$$\begin{aligned}
 \frac{d^2 t}{d\tau^2} &= 0 \\
 \frac{d^2 r}{d\tau^2} &= r \cdot \frac{d \varphi^2}{d\tau} + r \cdot \frac{d \theta^2}{d\tau} \\
 \frac{d^2 \theta}{d\tau^2} &= \frac{\sin(2\theta)}{2} \cdot \frac{d \varphi^2}{d\tau} - \frac{2 \, d \varphi \, d r}{r \, d\tau \, d\tau} \\
 \frac{d^2 \varphi}{d\tau^2} &= -\frac{2}{\tan(\theta)} \frac{d \varphi \, d \theta}{d\tau \, d\tau} - \frac{2 \, d \varphi \, d r}{r \, d\tau \, d\tau}
 \end{aligned} \tag{4.28}$$

Esto nos puede engañar un poco y a simple vista puede parecer que no sean líneas rectas como era de esperar. Sin embargo si vemos las geodésicas con las coordenadas cartesianas obtenemos líneas rectas cuando lo resolvemos:

$$\begin{aligned}\frac{d^2 t}{d\tau^2} &= 0 \\ \frac{d^2 x}{d\tau^2} &= 0 \\ \frac{d^2 y}{d\tau^2} &= 0 \\ \frac{d^2 z}{d\tau^2} &= 0\end{aligned}\tag{4.29}$$

Que una solución muy clara es:

$$\begin{aligned}t &= t_0 + \dot{t}_0 \cdot \tau \\ x &= x_0 + \dot{x}_0 \cdot \tau \\ y &= y_0 + \dot{y}_0 \cdot \tau \\ z &= z_0 + \dot{z}_0 \cdot \tau \\ \dot{t} &= \dot{t}_0 \\ \dot{x} &= \dot{x}_0 \\ \dot{y} &= \dot{y}_0 \\ \dot{z} &= \dot{z}_0\end{aligned}\tag{4.30}$$

Tratar de encontrar un solución para las ecuaciones de las geodésicas en coordenadas esféricas es mucho más complejo. Esto es un ejemplo de que como se mencionó previamente, los símbolos de Christoffel no son tensores porque no obedecen la propiedad 3.3.7 del cambio de cartas en tensores.

Ahora vamos a ver que esto no ocurre con las componentes del tensor curvatura de Ricci. Como se puede ver a simple vista las componentes del tensor de Ricci en coordenadas cartesianas claramente son todas nulas, mientras que no es tan claro hacer este cálculo con las coordenadas esféricas. Para calcular las componentes del tensor de Ricci en coordenadas esféricas usamos la función de Python desarrollada en este trabajo, como se puede ver en la figura 4.14, y obtenemos que todas las componentes del tensor de Ricci son nulas. A su vez también se puede ver que la curvatura escalar es 0, es decir, la métrica de Minkowski, como previamente hemos dicho, determina un espacio tiempo plano.

### 4.2.3. Métrica de Schwarzschild

La teoría de la relatividad general, postulada por Albert Einstein en sus artículos de 1915 y 1916 presentó la idea de la teoría de la relatividad especial. Esta teoría tiene una base puramente

### Componentes tensor de Ricci

```
In [14]: 1 Ricci_curv = componentes_tensor_ricci(g,variables)
2         for i in range(len(variables)):
3             for j in range(len(variables)):
4                 display(Math(f'R_{{{variables[i]},{variables[j]}}}} = '+latex(Ricci_curv[i][j]))
5
```

$$R_{t,t} = 0$$

$$R_{t,r} = 0$$

$$R_{t,theta} = 0$$

$$R_{t,phi} = 0$$

$$R_{r,t} = 0$$

$$R_{r,r} = 0$$

$$R_{r,theta} = 0$$

$$R_{r,phi} = 0$$

$$R_{theta,t} = 0$$

$$R_{theta,r} = 0$$

$$R_{theta,theta} = 0$$

$$R_{theta,phi} = 0$$

$$R_{phi,t} = 0$$

$$R_{phi,r} = 0$$

$$R_{phi,theta} = 0$$

$$R_{phi,phi} = 0$$

Figura 4.14: Salida de la función al calcular las componentes del tensor de Ricci con coordenadas esféricas.

geométrica. Estas bases se rigen por unas ecuaciones que describen el espacio, estas son las ecuaciones de campo de Einstein:

$$R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} + \Lambda g_{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu} \quad (4.31)$$

En estas ecuaciones se presentan las siguientes variables:

1.  $R_{\mu\nu}$  tensor de Ricci.
2.  $g_{\mu\nu}$  la componente  $\mu\nu$  de la métrica.
3.  $T_{\mu\nu}$  tensor energía momento

Además de las constante: la constante cosmológica,  $\Lambda$ , y la curvatura de Ricci,  $R$ . Como podemos ver la métrica tiene un papel crucial a la hora de determinar las ecuaciones de campo, por ello a toda métrica que lo cumpla se dirá que es una solución de las ecuaciones de campo de Einstein.

Existen diferentes soluciones para estas ecuaciones según cómo sea el espacio. Por ejemplo, la métrica de Minkowski sería la solución mas simple para estas ecuaciones. Una de las primeras soluciones sería la presentada por Karl Schwarzschild en 1916, unos pocos meses después de la presentación de los trabajos de Einstein. Esta solución presenta una distribución del universo con una masa esférica y constante. Esta métrica nos ayuda a teorizar cómo de extremas son las condiciones de un cuerpo en las cercanías de pulsares o de un agujeros negros.

Existen otras soluciones para a estas ecuaciones. Otra muy similar a la solución de Schwarzschild con una masa en el centro pero con una simetría axial y en rotación sería presentada casi 50 años después por el físico Roy Patrick Kerr. O la famosa solución de Friedman-Roberson-Walker de un universo homogéneo e isótropo.

Encontrar soluciones para las ecuaciones de campo de Einstein también ha sido un pasatiempo para grandes matemáticos como pueden ser Kurt Gödel que propuso una solución para un universo homogéneo y en rotación.[7]

En este trabajo nos vamos a centrar en la solución de Schwarzschild, ya que fue el tema inicial de este trabajo. La métrica de Schwarzschild expresada en coordenadas esféricas es de la siguiente forma:

$$g = -c^2 \left( 1 - \frac{2GM}{c^2 r} \right) dt \otimes dt + \frac{1}{\left( 1 - \frac{2GM}{c^2 r} \right)} dr \otimes dr + r^2 ( d\theta \otimes d\theta + \sin^2 \theta d\varphi \otimes d\varphi ) \quad (4.32)$$

Las constantes que se pueden observar en esta métrica son:

- $c$ , velocidad de la luz en el vacío,  $3 \times 10^8 \frac{m}{s}$

- G, constante de gravitación universal,  $6,67 \times 10^{-11} \frac{Nm^2}{kg^2}$
- M, masa del objeto que perturba el espacio tiempo.

Las variables que hay en esta métrica:

- $r$ , distancia a la que se encuentra el objeto del origen.
- $\theta$ , es la colatitud<sup>1</sup>.
- $\varphi$ , es el ángulo azimutal<sup>2</sup>.

En esta métrica se puede observar un parámetro muy importante  $\frac{2GM}{c^2}$ . Esta expresión representa el radio de Schwarzschild o radio del horizonte de sucesos, es decir el radio de un agujero negro con una masa M. Por ejemplo, la masa del sol es aproximadamente  $2 \cdot 10^{30}$  entonces el radio de un agujero negro del tamaño del sol sería aproximadamente de 3000 metros. [4]

Se puede calcular que en esa métrica hay dos singularidades, una debida a las cartas usadas y otra física:

- $r = \frac{2GM}{c^2}$  a esta singularidad es llamada *radio de Schwarzschild* u horizonte de sucesos y es el radio del agujero negro. Esta singularidad es no alcanzable, como se demuestra en la proposición 4.2.2. Además, se puede ver como la signatura de la métrica cambia cuando  $r < \frac{2GM}{c^2}$  de  $(-, +, +, +)$  a  $(+, -, +, +)$ , lo cual nos indica que la métrica no se comporta bien dentro del radio de Schwarzschild.
- $r = 0$  esta singularidad es la del agujero negro.

Se puede apreciar que si la masa es nula o el radio es mucho mayor que el radio de Schwarzschild entonces la métrica se convierte en la métrica de Minkowski en coordenadas esféricas:

- Si tomamos  $M = 0$ :

$$\begin{aligned}
 g &= -c^2 \left( 1 - \frac{2G0}{c^2 r} \right) dt \otimes dt + \frac{1}{\left( 1 - \frac{2G0}{c^2 r} \right)} dr \otimes dr + r^2 ( d\theta \otimes d\theta + \sin^2 \theta d\phi \otimes d\phi) \\
 g &= -c^2 dt \otimes dt + dr \otimes dr + r^2 ( d\theta \otimes d\theta + \sin^2 \theta d\phi \otimes d\phi)
 \end{aligned}
 \tag{4.33}$$

<sup>1</sup>Opuesto del ángulo que hay a uno de los polos

<sup>2</sup>Ángulo que con el meridiano forma el círculo vertical que pasa por un punto de la esfera celeste o del globo terráqueo.

- Si  $r \gg \frac{2GM}{c^2}$  entonces  $\frac{2GM}{c^2 r} \approx 0$ :

$$g = -c^2 (1 - 0) dt \otimes dt + \frac{1}{(1 - 0)} dr \otimes dr + r^2 (d\theta \otimes d\theta + \sin^2 \theta d\phi \otimes d\phi) \quad (4.34)$$

Volvemos a obtener la métrica de Minkowski en coordenadas esféricas, (4.25).

Primero vamos a ver los símbolos de Christoffel de esta métrica, para ello utilizaremos la función desarrollada para ello, como se puede observar en la figura 4.15.

```

3 #Esta matriz va a tener los simbolos de chistoffle
4 #Por ejemplo, simbolo_chirtoffle[2][0][1] seria \gamma^y_{t,x}
5 simbolos_chistoffle = [[[0 for j in variables] for i in variables] for m in variables]
6
7
8 for m in range(len(variables)):
9     for i in range(len(variables)):
10        for j in range(len(variables)):
11            simbolo = 0
12            #El sumatorio respecto del indice libre l
13            for l in range(len(variables)):
14                #Para evitar hacer una multiplicación por 0
15                if g_inv[m][l] != 0:
16                    simbolo = simbolo + g_inv[m][l]*(derivadas_variable[j][i][l]+derivadas_variable[i][j][l]-deri
17                #Aquí multiplico por el 1/2 que hemos sacado de factor común
18                simbolo = 1/2*simbolo
19            simbolos_chistoffle[m][i][j] = simplify(simbolo)
20 for i in range(len(simbolos_chistoffle)):
21     for j in range(len(simbolos_chistoffle)):
22         for k in range(len(simbolos_chistoffle)):
23             if simbolos_chistoffle[i][j][k] != 0:
24
25                 display(Math(f'\Gamma^{{{variables[i]}}}_{{{variables[j]},{variables[k]}}} = ' + latex(simbolos_c

```

$$\Gamma_{r,r}^r = \frac{0.5r_s}{r(r-r_s)}$$

$$\Gamma_{r,r}^r = \frac{0.5r_s}{r(r-r_s)}$$

$$\Gamma_{r,t}^r = \frac{0.5c^2 r_s (r-r_s)}{r^3}$$

$$\Gamma_{r,r}^r = -\frac{0.5r_s}{r(r-r_s)}$$

$$\Gamma_{\theta,\theta}^r = -1.0r + 1.0r_s$$

$$\Gamma_{\phi,\phi}^r = 1.0(-r + r_s) \sin^2(\theta)$$

$$\Gamma_{r,\theta}^{\theta} = \frac{1.0}{r}$$

$$\Gamma_{\theta,r}^{\theta} = \frac{1.0}{r}$$

$$\Gamma_{\phi,\phi}^{\theta} = -0.5 \sin(2\theta)$$

$$\Gamma_{r,\phi}^{\phi} = \frac{1.0}{r}$$

$$\Gamma_{\theta,\phi}^{\phi} = \frac{1.0}{\tan(\theta)}$$

$$\Gamma_{\phi,r}^{\phi} = \frac{1.0}{r}$$

$$\Gamma_{\phi,\theta}^{\phi} = \frac{1.0}{\tan(\theta)}$$

Figura 4.15: Símbolos de Christoffel de la métrica de Schwarzschild en coordenadas esféricas



Las ecuaciones de las geodésicas con las coordenadas esféricas son:

$$\begin{aligned}
\frac{d^2 t}{d\lambda^2} &= -\frac{2GM}{r(2GM - c^2 r)} \frac{dr}{d\lambda} \frac{dt}{d\lambda} \\
\frac{d^2 r}{d\lambda^2} &= -\frac{GM}{r(2GM - c^2 r)} \frac{dr^2}{d\lambda} + \frac{GM(2GM - c^2 r)}{c^2 r^3} \frac{dt^2}{d\lambda} \\
&\quad + \frac{(-2GM + c^2 r) \frac{d\theta^2}{d\lambda}}{c^2} + \frac{(-2GM + c^2 r) \frac{d\varphi^2}{d\lambda} \sin^2 \theta}{c^2} \\
\frac{d^2 \theta}{d\lambda^2} &= -\frac{1}{2} \frac{d\varphi^2}{d\lambda} \sin(2\theta) - \frac{2}{r} \frac{dr}{d\lambda} \frac{d\theta}{d\lambda} \\
\frac{d^2 \varphi}{d\lambda^2} &= -\frac{2}{\tan(\theta)} \frac{d\varphi}{d\lambda} \frac{d\theta}{d\lambda} - \frac{2}{r} \frac{d\varphi}{d\lambda} \frac{dr}{d\lambda}
\end{aligned} \tag{4.35}$$

El parámetro  $\lambda$  del que depende las funciones será  $\tau$  cuando la geodésica sea temporal, en cualquier otro caso será diferente.

Ahora vamos a resolver las ecuaciones. Para conseguir este objetivo tomaremos una serie de simplificaciones. La primera será en cuanto a la notación:

$$\frac{dt}{d\lambda} = u^0, \quad \frac{dr}{d\lambda} = u^1, \quad \frac{d\theta}{d\lambda} = u^2 \quad y \quad \frac{d\varphi}{d\lambda} = u^3 \tag{4.36}$$

Tomamos las condiciones iniciales:

$$\begin{aligned}
t(0) &= 0 & \left. \frac{dt}{d\lambda} \right|_{\lambda=0} &= u_0^0 \\
r(0) &= r_0 & \left. \frac{dr}{d\lambda} \right|_{\lambda=0} &= u_0^1 \\
\theta(0) &= \frac{\pi}{2} & \left. \frac{d\theta}{d\lambda} \right|_{\lambda=0} &= 0 \\
\varphi(0) &= 0 & \left. \frac{d\varphi}{d\lambda} \right|_{\lambda=0} &= u_0^3
\end{aligned} \tag{4.37}$$

Obteniendo las ecuaciones de las geodésicas queda de la siguiente forma:

$$\frac{du^0}{d\lambda} = -\frac{2GM u^1 u^0}{r(2GM - c^2 r)} \tag{4.38a}$$

$$\begin{aligned}
\frac{du^1}{d\lambda} &= -\frac{GM}{r(2GM - c^2 r)} (u^1)^2 + \frac{GM(2GM - c^2 r)}{c^2 r^3} (u^0)^2 \\
&\quad + \frac{(-2GM + c^2 r)}{c^2} (u^3)^2
\end{aligned} \tag{4.38b}$$

$$\frac{du^3}{d\lambda} = -\frac{2u^3 u^1}{r} \tag{4.38c}$$

Observamos que las ecuaciones (4.38a) y (4.38c) son ecuaciones diferenciales lineales de variables separables si deshacemos el cambio de  $\frac{dr}{d\lambda} = u^1$ .

$$\frac{1}{u^0} \frac{du^0}{d\lambda} = -\frac{2GM}{r(2GM - c^2r)} \frac{dr}{d\lambda} \quad (4.39a)$$

$$\frac{1}{u^3} \frac{du^3}{d\lambda} = \frac{-2}{r} \frac{dr}{d\lambda} \quad (4.39b)$$

Entonces integramos ambas ecuaciones diferenciales:

$$\begin{aligned} \int_{u_0^0}^{u^0} \frac{1}{u^0} du^0 &= \int_{r_0}^r -\frac{2GM}{r(2GM - c^2r)} dr \\ \log(u^0) \Big|_{u_0^0}^{u^0} &= \log \left( \frac{r - \frac{2GM}{c^2}}{r} \right) \Big|_{r_0}^r \end{aligned} \quad (4.40a)$$

$$\begin{aligned} \int_{u_0^3}^{u^3} \frac{1}{u^3} du^3 &= \int_{r_0}^r \frac{-2}{r} dr \\ \log(u^3) \Big|_{u_0^3}^{u^3} &= \log \left( \frac{1}{r^2} \right) \Big|_{r_0}^r \end{aligned} \quad (4.40b)$$

Obteniendo que:

$$\begin{aligned} u^0 &= u_0^0 \cdot \left( \frac{r(r_0 - \frac{2GM}{c^2})}{r_0(r - \frac{2GM}{c^2})} \right) \\ u^3 &= \frac{u_0^3 r_0^2}{r^2} \end{aligned} \quad (4.41)$$

Si recordamos la proposición 3.6.11, por ser una geodésica la curva tendrá módulo constante, lo denotaremos por  $k$ . Entonces, podemos utilizar las simplificaciones previamente mencionadas<sup>3</sup> en la ecuación (4.32) y la propiedad 3.6.11:

$$-c^2 \left(1 - \frac{r_s}{r}\right) (u^0)^2 + \frac{1}{\left(1 - \frac{r_s}{r}\right)} (u^1)^2 + r^2 (u^3)^2 = k \quad (4.42)$$

Sustituyendo las ecuaciones (4.41) en la ecuación (4.42) obtenemos que:

$$-c^2 \left(1 - \frac{r_s}{r}\right) \left(u_0^0 \cdot \left(\frac{r(r_0 - r_s)}{r_0(r - r_s)}\right)\right)^2 + \frac{1}{\left(1 - \frac{r_s}{r}\right)} (u^1)^2 + r^2 \left(\frac{u_0^3 r_0^2}{r^2}\right)^2 = k \quad (4.43)$$

---

<sup>3</sup>Añadiendo la simplificación de notación  $r_s = \frac{2GM}{c^2}$

Despejamos  $u^1$ :

$$\begin{aligned}
\frac{1}{\left(1 - \frac{r_s}{r}\right)} (u^1)^2 &= k + c^2 \left(1 - \frac{r_s}{r}\right) \left(u_0^0 \cdot \left(\frac{r(r_0 - r_s)}{r_0(r - r_s)}\right)\right)^2 \\
&\quad - r^2 \left(\frac{u_0^3 r_0^2}{r^2}\right)^2 \\
(u^1)^2 &= k \cdot \left(1 - \frac{r_s}{r}\right) + -r^2 \left(\frac{u_0^3 r_0^2}{r^2}\right)^2 \cdot \left(1 - \frac{r_s}{r}\right) \\
&\quad c^2 \left(1 - \frac{r_s}{r}\right)^2 \left(u_0^0 \cdot \left(\frac{r(r_0 - r_s)}{r_0(r - r_s)}\right)\right)^2 \tag{4.44}
\end{aligned}$$

Simplificamos el lado derecho de la igualdad

$$\begin{aligned}
&= k \cdot \frac{r - r_s}{r} + c^2 (u_0^0)^2 \left(\frac{(r_0 - r_s)^2}{r_0^2}\right) - \\
&\quad (u_0^3 r_0^2)^2 \cdot \left(\frac{r - r_s}{r^3}\right)
\end{aligned}$$

Finalmente obtenemos que:

$$u^1 = \sqrt{k \cdot \frac{r - r_s}{r} + c^2 (u_0^0)^2 \left(\frac{(r_0 - r_s)^2}{r_0^2}\right) - (u_0^3 r_0^2)^2 \cdot \left(\frac{r - r_s}{r^3}\right)} \tag{4.45}$$

Concluimos que la ecuación de las geodésicas en la métrica de Schwarzschild en coordenadas esféricas, deshaciendo la simplificación de la notación previamente mencionada, es:

$$\begin{aligned}
\frac{dt}{d\lambda} &= \frac{dt}{d\lambda} \Big|_{\lambda=0} \cdot \left(\frac{r_0(r - \frac{2GM}{c^2})}{r(r_0 - \frac{2GM}{c^2})}\right) \\
\frac{dr}{d\lambda} &= \sqrt{k \cdot \frac{r - r_s}{r} + c^2 \left(\frac{dt}{d\lambda} \Big|_{\lambda=0}\right)^2 \left(\frac{(r_0 - r_s)^2}{r_0^2}\right) - \left(\frac{d\varphi}{d\lambda} \Big|_{\lambda=0} \cdot r_0^2\right)^2 \cdot \left(\frac{r - r_s}{r^3}\right)} \\
\frac{d\varphi}{d\lambda} &= \frac{d\varphi}{d\lambda} \Big|_{\lambda=0} \frac{r_0^2}{r^2} \tag{4.46}
\end{aligned}$$

**Proposición 4.2.2.** [Ver [4]] Sea  $\alpha$  una geodésica que se dirige hacia el punto  $(0, 0, 0)$  entonces solamente alcanzará  $r = \frac{2GM}{c^2}$  en un tiempo infinito.

*Demostración.* Para demostrar esto vamos a hacer la simplificación de la métrica tomando los valores iniciales  $v = (0, r_0, \frac{\pi}{2}, 0)$   $\vec{v} = (\dot{t}, \dot{r}, 0, 0)$  con  $r_0 > r_s$  y  $\dot{r} < 0$ . A esta simplificación de la métrica de Schwarzschild se le llama métrica radial y su expresión es:

$$g = -c^2 \left(1 - \frac{2GM}{c^2 r}\right) dt \otimes dt + \frac{1}{\left(1 - \frac{2GM}{c^2 r}\right)} dr \otimes dr \tag{4.47}$$

Las ecuaciones de las geodésicas en esta métrica son:

$$\frac{d^2t}{d\lambda^2} = -\frac{r_s \frac{dr}{d\lambda} \frac{dt}{d\lambda}}{r(r_s - r)} \quad (4.48a)$$

$$\frac{d^2r}{d\lambda^2} = -\frac{r_s \frac{dr^2}{d\lambda}}{r(r_s - r)} + \frac{r_s \frac{dt^2}{d\lambda}}{2r \cdot (r - r_s)} \quad (4.48b)$$

Por simplificar la notación se va a usar esta terminología:

$$\frac{dt}{d\lambda} = u^0 \quad \frac{dr}{d\lambda} = u^1 \quad (4.49)$$

Vamos a obtener una expresión de  $u^0$  en función del radio operando sobre la ecuación (4.48a):

$$\begin{aligned} \frac{1}{u^0} \frac{du^0}{d\lambda} + \frac{r_s}{r(r_s - r)} u^1 &= 0 \\ \frac{1}{u^0} \frac{du^0}{d\lambda} - \frac{u^1}{r} + \frac{u^1}{r - r_s} &= 0 \\ \frac{d(\log(u^0))}{d\lambda} - \frac{d(\log(r))}{d\lambda} + \frac{d(\log(r - r_s))}{d\lambda} &= 0 \\ \frac{d\left(\log\left(\frac{(r - r_s)u^0}{r}\right)\right)}{d\lambda} &= 0 \end{aligned} \quad (4.50)$$

Por lo tanto obtenemos que:

$$\frac{(r - r_s)u^0}{r} = e^b$$

La expresión buscada es:

$$u^0 = \frac{e^b \cdot r}{r - r_s} \quad (4.51)$$

Para simplificar la notación utilizaremos  $B$  para denotar  $e^b$ . Ahora vamos a utilizar la propiedad 3.6.11 como se hizo en el cálculo de las geodésicas:

$$-c^2 \left(\frac{r - r_s}{r}\right) (u^0)^2 + \frac{1}{\left(\frac{r - r_s}{r}\right)} (u^1)^2 = k \quad (4.52)$$

Sustituimos  $\frac{u^0}{B} = \frac{r}{r - r_s}$  en la ecuación anterior:

$$\begin{aligned} -c^2 \left(\frac{B}{u^0}\right) (u^0)^2 + \frac{u^0}{B} (u^1)^2 &= k \\ -c^2 B^2 u^0 + u^0 (u^1)^2 &= Bk \\ (u^1)^2 &= \frac{Bk}{u^0} + c^2 B^2 \end{aligned} \quad (4.53)$$

Hacemos la razón entre  $u^1$  y  $u^0$ :

$$\begin{aligned}
\left(\frac{u^0}{u^1}\right)^2 &= \frac{(u^0)^2}{\frac{Bk}{u^0} + c^2 B^2} \\
&= \frac{(u^0)^3}{Bk + u^0 c^2 B^2} \\
&= \frac{(u^0)^3}{Bk + u^0 c^2 B^2} \\
&= \frac{(u^0)^2}{c^2 B^2} - \frac{\frac{k}{c^2 B}}{Bk + c^2 B^2 u^0} \\
&= \frac{(u^0)^2}{c^2 B^2} - \frac{k}{c^2 k + c^4 B u^0} \\
&= \frac{1}{c^2 B^2} B^2 \left(\frac{r}{r-r_s}\right)^2 - \frac{k}{c^2 k + c^4 B \frac{B \cdot r}{r-r_s}} \\
&= \frac{1}{c^2} \left(\frac{r}{r-r_s}\right)^2 - \frac{k(r-r_s)}{c^2 k(r-r_s) + c^4 B^2 r}
\end{aligned} \tag{4.54}$$

Como  $\frac{u^0}{u^1} = \frac{dt}{dr} < 0$  podemos concluir que

$$\frac{dt}{dr} = - \left( \frac{1}{c^2} \left(\frac{r}{r-r_s}\right)^2 - \frac{k(r-r_s)}{c^2 k(r-r_s) + c^4 B^2 r} \right)^{\frac{1}{2}}.$$

Integrando la ecuación anterior tenemos que

$$\begin{aligned}
t(r) &= t(r_0) + \int_r^{r_0} \left( \frac{1}{c^2} \left(\frac{x}{x-r_s}\right)^2 - \frac{k(x-r_s)}{c^2 k(x-r_s) + c^4 B^2 r} \right)^{\frac{1}{2}} dx \\
&\geq t(r_0) + \frac{1}{c} \int_r^{r_0} \frac{x}{x-r_s} dx \\
&= t(r_0) + \frac{1}{c} \left( r - r_0 + \log \left( \frac{r_0 - r_s}{r - r_s} \right) \right)
\end{aligned}$$

Donde hemos aplicado que el segundo sumando dentro de la raíz de la primera integral va a ser siempre positivo, ya que  $k \leq 0$ . Por lo que finalmente

$$\lim_{r \rightarrow r_s^+} t(r) = +\infty.$$

Además, esta propiedad se puede observar también resolviendo el sistema de forma numérica, como se puede ver en la figura 4.16, como tiende asintóticamente al radio de Schwarzschild.

Es decir que tarda un tiempo de Schwarzschild infinito en alcanzar el radio de Schwarzschild pero sólo necesita un tiempo propio finito para alcanzar este radio. A este fenómeno se le conoce como la **censura cósmica**.

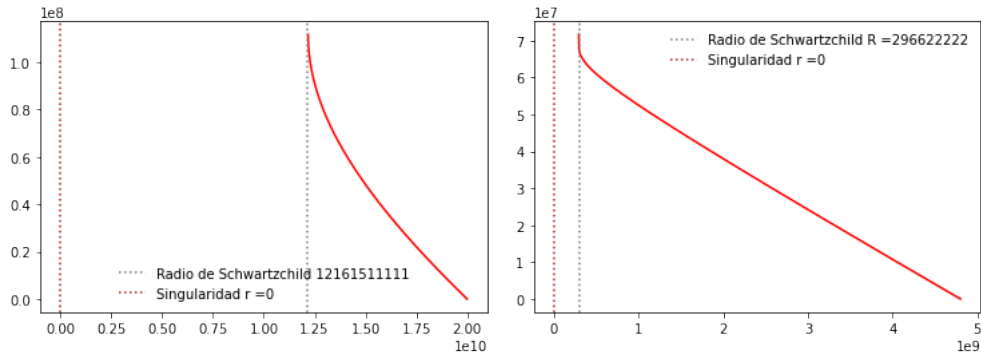


Figura 4.16: Variación del radio de la geodésica a lo largo del tiempo con un agujero negro de  $8,2 \times 10^{35}$  y  $2 \times 10^{36}$  respectivamente

Se resolvieron las ecuaciones de las geodésicas de manera numérica con la implementación de Runge-Kutta, que se puede observar en el apéndice C.

Algunos de los resultados obtenidos son:

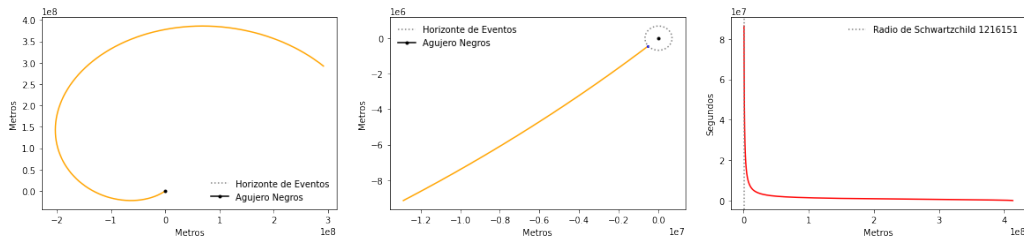


Figura 4.17: Caída de un fotón en un agujero negro de masa  $8,2 \times 10^{32}$

Entonces podemos dividir el espacio en dos zonas: cuando  $0 < r < \frac{2GM}{c^2}$  y cuando  $\frac{2GM}{c^2} < r$ , es decir, que el espacio no es completo. A nosotros nos interesa tener todo el espacio no este dividido. Ante esta problemática cuando el radio de la partícula es igual que el radio del agujero negro surge la solución de un cambio de cartas. Pero esta solución aparentemente tan simple como un cambio de cartas no ocurrió hasta que Eddington (1924) las ideó y Finklestein (1958) las redescubrió y se presentaron las coordenadas que permitían ver juntas estas dos regiones del espacio tiempo, pero presentaban algunos problemas con las geodésicas salientes, hasta los inicios década de los años 60 cuando Kruskal y Szekeres resolvieron este problema para todas las geodésicas.

Ahora vamos a ver la curvatura de la métrica de Schwarzschild. Las componentes del tensor de Riemann calculadas con la función desarrollada para ello, como se puede observar en la figura 4.19 .

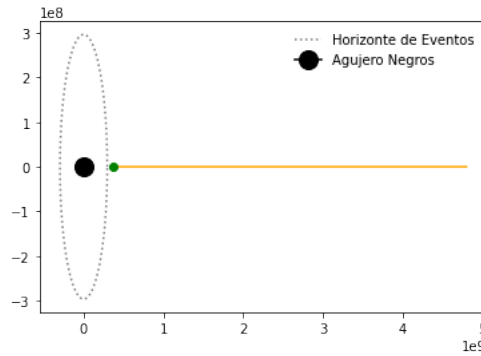


Figura 4.18: Gráfica de una curva geodésica con la métrica radial

En ellas vamos a observar que se mantienen las dos singularidades que se han mencionado antes, cuando  $R=0$  y cuando  $R=\frac{2GM}{c^2}$ . Pero como vemos muchas de las componentes del tensor de Riemann no aportan ninguna información. Como previamente hemos dicho el tensor de Riemann y el tensor de Ricci aportan la misma información para variedades de hasta dimensión 3, pero esto no es cierto para dimensiones superiores. Si calculamos las componentes del tensor de Ricci, como se puede ver en la figura 4.20, todas las componentes del tensor de Ricci son nulas y por lo tanto aparentemente nos diría que es una variedad plana sin ninguna perturbación. Además, la curvatura escalar también será nula.

### Espacio tiempo de Schwarzschild en coordenadas Cartesianas isotrópicas

El objetivo de usar un la métrica de Schwarzschild en coordenadas isotrópicas es que se parezca lo máximo posible a un espacio euclídeo, por eso se va a utilizar también las unidades del sistema métrico internacional y no las unidades geometrizadas<sup>4</sup>. [16][15]

Entonces si recordamos en el ejemplo 3.5.6 la métrica euclídea de  $\mathbb{R}^3$  en coordenadas esféricas es de la forma:

$$g_e = dr \otimes dr + r^2 d\theta \otimes d\theta + r^2 \sin(\theta)^2 d\phi \otimes d\phi \quad (4.55)$$

Entonces se puede expresar la métrica de Schwarzschild como:

$$g_s = -c^2 A^2(r) dt \otimes dt + B^2(r) \cdot g_e \quad (4.56)$$

Buscar unas funciones  $A(r)$  y  $B(r)$  que hagan que para todo punto de  $p \in \mathbb{R}^4$  sea isotrópico.

<sup>4</sup>Las unidades geometrizadas utilizan  $G=c=1$  para simplificar la notación.

### Tensor curvatura de Riemann

In [10]: 1 R = componentes\_tensor\_riemann(g, variables)

```
In [11]: 1 for l in range(len(variables)):
2         for i in range(len(variables)):
3             for j in range(len(variables)):
4                 for k in range(len(variables)):
5                     if R[l][i][j][k] != 0:
6                         display(Math(f'R^{{{{variables[l]}}}_{{{variables[i]},{variables[j]},{variables[k]}}}} =
```

$$R^l_{r,r,r} = \frac{2.0GM}{r^2 (-2.0GM + 1.0c^2r)}$$

$$R^l_{t,theta,theta} = -\frac{1.0GM}{c^2r}$$

$$R^l_{t,phi,phi} = -\frac{1.0GM \sin^2(\theta)}{c^2r}$$

$$R^l_{r,t,r} = \frac{2.0GM}{r^2 (2.0GM - 1.0c^2r)}$$

$$R^l_{theta,t,theta} = \frac{1.0GM}{c^2r}$$

$$R^l_{phi,t,phi} = \frac{1.0GM \sin^2(\theta)}{c^2r}$$

$$R^r_{r,t,t} = -\frac{GM (4.0GM - 2.0c^2r)}{c^2r^4}$$

$$R^r_{r,t,t} = \frac{GM (4.0GM - 2.0c^2r)}{c^2r^4}$$

$$R^r_{r,theta,theta} = -\frac{1.0GM}{c^2r}$$

$$R^r_{theta,r,theta} = \frac{1.0GM}{c^2r}$$

$$R^r_{phi,r,phi} = \frac{1.0GM \sin^2(\theta)}{c^2r}$$

$$R^{theta}_{t,theta,t} = \frac{1.0GM (2GM - c^2r)}{c^2r^4}$$

$$R^{theta}_{r,theta,r} = -\frac{1.0GM}{r^2 (2GM - c^2r)}$$

$$R^{theta}_{theta,t,t} = -\frac{1.0GM (2GM - c^2r)}{c^2r^4}$$

$$R^{theta}_{theta,r,r} = \frac{1.0GM}{r^2 (2GM - c^2r)}$$

$$R^{theta}_{theta,phi,phi} = \frac{2.0GM \sin^2(\theta)}{c^2r}$$

$$R^{theta}_{phi,theta,phi} = -\frac{2.0GM \sin^2(\theta)}{c^2r}$$

$$R^{phi}_{t,phi,t} = \frac{1.0GM (2GM - c^2r)}{c^2r^4}$$

$$R^{phi}_{r,phi,r} = -\frac{1.0GM}{r^2 (2GM - c^2r)}$$

$$R^{phi}_{theta,phi,theta} = -\frac{2.0GM}{c^2r}$$

$$R^{phi}_{phi,t,t} = -\frac{1.0GM (2GM - c^2r)}{c^2r^4}$$

$$R^{phi}_{phi,r,r} = \frac{1.0GM}{r^2 (2GM - c^2r)}$$

$$R^{phi}_{phi,theta,theta} = \frac{2.0GM}{c^2r}$$

Figura 4.19: Cálculo de las componentes del tensor de Riemann con la métrica de Schwarzschild.

Esas dos funciones son:

$$A(r) = \frac{1 - \frac{r_s}{r}}{1 + \frac{r_s}{r}} \tag{4.57}$$

$$B(r) = \left(1 + \frac{r_s}{r}\right)^2$$



### Tensor curvatura de Ricci

```
In [7]: 1 Ricci_curv = componentes_tensor_ricci(g,variables)
2 for i in range(len(variables)):
3     for j in range(len(variables)):
4         #if Ricci_curv[i][j] != 0:
5             display(Math(f'R_{{{variables[i]},{variables[j]}}}} = '+latex(sp.simplify(Ricci_curv[i][j]))))

R_{t,t} = 0
R_{t,r} = 0
R_{t,theta} = 0
R_{t,phi} = 0
R_{r,t} = 0
R_{r,r} = 0
R_{r,theta} = 0
R_{r,phi} = 0
R_{theta,t} = 0
R_{theta,r} = 0
R_{theta,theta} = 0
R_{theta,phi} = 0
R_{phi,t} = 0
R_{phi,r} = 0
R_{phi,theta} = 0
R_{phi,phi} = 0
```

Figura 4.20: Componentes del tensor de Ricci de la métrica de Schwarzschild

Entonces, la métrica de Schwarzschild expresada como en la ecuación (4.56) y las funciones anteriores: [16]

$$g = - \left( \frac{1 - r_s/r}{1 + r_s/r} \right)^2 c^2 dt \otimes dt + \left( 1 + \frac{r_s}{r} \right)^4 [dr \otimes dr + r^2 (d\theta \otimes d\theta + \sin^2 \theta d\varphi \otimes d\varphi)] \quad (4.58)$$

$$r_s = \frac{2GM}{c^2}$$

Para poder trabar de una manera más simple e intuitiva se utilizaron las coordenadas cartesianas:

$$\begin{aligned} t &= t \\ r &= \sqrt{x^2 + y^2 + z^2} \\ \theta &= \arccos\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right) \\ \varphi &= \arctan\left(\frac{y}{x}\right) \end{aligned} \quad (4.59)$$

Obteniendo:

$$\begin{aligned}
dt &= dt \\
dr &= \frac{xdx}{\sqrt{x^2 + y^2 + z^2}} + \frac{ydy}{\sqrt{x^2 + y^2 + z^2}} + \frac{zdz}{\sqrt{x^2 + y^2 + z^2}} \\
d\theta &= \frac{xzdx}{\sqrt{\frac{x^2+y^2}{x^2+y^2+z^2}} (x^2 + y^2 + z^2)^{3/2}} + \frac{yzdy}{\sqrt{\frac{x^2+y^2}{x^2+y^2+z^2}} (x^2 + y^2 + z^2)^{3/2}} \\
&\quad + \frac{(-x^2 - y^2) dz}{\sqrt{\frac{x^2+y^2}{x^2+y^2+z^2}} (x^2 + y^2 + z^2)^{3/2}} \\
d\varphi &= \frac{-y}{x^2 + y^2} dy + \frac{y}{x^2 + y^2} dx
\end{aligned} \tag{4.60}$$

Finalmente sustituimos los resultados de la ecuación (4.60) en la métrica de Schwarzschild isotrópica, (4.58), y obtenemos que:

$$\begin{aligned}
g &= - \left( \frac{1 - \frac{\rho_s}{\rho}}{1 + \frac{\rho_s}{\rho}} \right) c^2 dt \otimes dt + \left( 1 + \frac{\rho_s}{\rho} \right)^4 \left[ dx \otimes dx + dy \otimes dy + dz \otimes dz \right] \\
\rho_s &= \frac{GM}{4c^2} \\
\rho^2 &= x^2 + y^2 + z^2.
\end{aligned} \tag{4.61}$$

Su matriz asociada es:

$$\begin{bmatrix}
-\frac{c^2 \left( 1 - \frac{\rho_s(M)}{\sqrt{x^2+y^2+z^2}} \right)^2}{\left( 1 + \frac{\rho_s(M)}{\sqrt{x^2+y^2+z^2}} \right)^2} & 0 & 0 & 0 \\
0 & \left( 1 + \frac{\rho_s(M)}{\sqrt{x^2+y^2+z^2}} \right)^4 & 0 & 0 \\
0 & 0 & \left( 1 + \frac{\rho_s(M)}{\sqrt{x^2+y^2+z^2}} \right)^4 & 0 \\
0 & 0 & 0 & \left( 1 + \frac{\rho_s(M)}{\sqrt{x^2+y^2+z^2}} \right)^4
\end{bmatrix} \tag{4.62}$$

Esta métrica presenta las mismas singularidades que la métrica de Schwarzschild pero en diferente posición, es decir presentará una singularidad cuando el radio sea igual a  $\frac{GM}{4c^2}$  y cuando sea nulo. Entonces introducimos este métrica en nuestro programa y obtenemos que los símbolos de Christoffel son:

$$\begin{aligned}
\Gamma_{tt}^x &= \frac{2c^2 \rho^3 \rho_s (\rho - \rho_s) x}{(\rho + \rho_s)^7}, & \Gamma_{tt}^y &= \frac{2c^2 \rho^3 \rho_s (\rho - \rho_s) y}{(\rho + \rho_s)^7}, & \Gamma_{tt}^z &= \frac{2c^2 \rho^3 \rho_s (\rho - \rho_s) z}{(\rho + \rho_s)^7} \\
\Gamma_{tx}^t &= \frac{2\rho_s x}{\rho^3 [1 - \rho_s^2/\rho^2]}, & \Gamma_{ty}^t &= \frac{2\rho_s y}{\rho^3 [1 - \rho_s^2/\rho^2]}, & \Gamma_{tz}^t &= \frac{2\rho_s z}{\rho^3 [1 - \rho_s^2/\rho^2]} \\
\Gamma_{xx}^x &= \Gamma_{xy}^x = \Gamma_{xz}^x = -\Gamma_{yy}^x = -\Gamma_{zz}^x = -\frac{2\rho_s}{\rho^3} \frac{x}{1 + \rho_s/\rho} \\
\Gamma_{xx}^y &= -\Gamma_{xy}^x = -\Gamma_{yy}^y = -\Gamma_{yz}^y = \Gamma_{zz}^y = \frac{2\rho_s}{\rho^3} \frac{y}{1 + \rho_s/\rho} \\
\Gamma_{xx}^z &= -\Gamma_{xz}^x = \Gamma_{yy}^z = -\Gamma_{yz}^y = -\Gamma_{zz}^z = \frac{2\rho_s}{\rho^3} \frac{z}{1 + \rho_s/\rho}
\end{aligned} \tag{4.63}$$

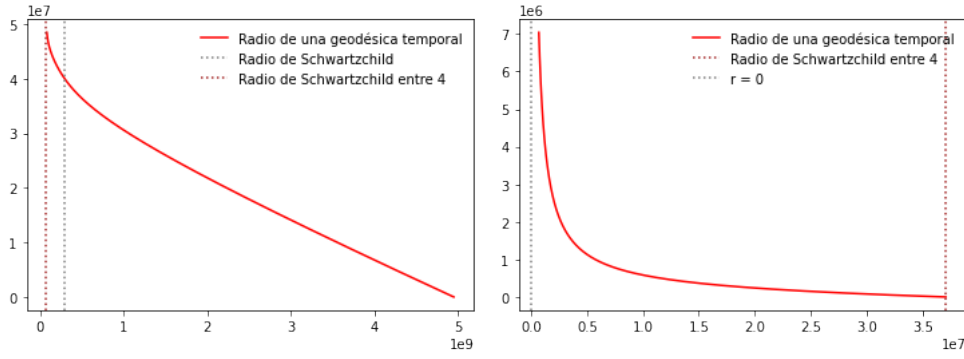


Figura 4.21: Ejemplo comportamiento de un geodésica espacial entorno a un agujero con una masa de  $2 \times 10^{35} Kg$  si está dentro o fuera del radio.

Y la ecuación de las geodésicas tomando la simplificación  $z = \dot{z} = 0$  es:

$$\begin{aligned}
 \frac{d^2 t}{d\tau^2} &= -4 \frac{4\rho_s \frac{dt}{d\lambda}}{\rho(\rho^2 - \rho_s^2)} \left[ \frac{dx}{d\lambda} x + \frac{dy}{d\lambda} y \right] \\
 \frac{d^2 x}{d\tau^2} &= -2 \frac{c^2 \rho_s (\rho_s - \rho) \rho^{\frac{3}{2}} x}{(\rho_s + \rho)^7} \frac{dt^2}{d\lambda} - \frac{2\rho_s}{(\rho_s + \rho)\rho^2} \left( x \left( \frac{dy^2}{d\lambda} - \frac{dx^2}{d\lambda} \right) - 2y \frac{dy}{d\lambda} \frac{dx}{d\lambda} \right) \\
 \frac{d^2 y}{d\tau^2} &= -2 \frac{c^2 \rho_s (\rho_s - \rho) \rho^{\frac{3}{2}} y}{(\rho_s + \rho)^7} \frac{dt^2}{d\lambda} - \frac{2\rho_s}{(\rho_s + \rho)\rho^2} \left( y \left( \frac{dx^2}{d\lambda} - \frac{dy^2}{d\lambda} \right) - 2x \frac{dy}{d\lambda} \frac{dx}{d\lambda} \right)
 \end{aligned} \tag{4.64}$$

Entonces se intentó solucionar de forma analítica ya que las ecuaciones eran muy similares pero no se obtuvo ningún resultado esclarecedor. Por lo tanto se utilizó el método de Runge-Kutta para poder encontrar las soluciones, como se puede ver en la figura . Se puede apreciar claramente como la geodésica es capaz de atravesar el horizonte de sucesos pero los datos obtenidos no mantienen la norma, cómo se mencionó en la proposición 4.2.2 las curvas geodésicas tienen módulo constante. Esto nos indica que, como se menciona en [4], la métrica de Schwarzschild no es una métrica estática y se produce un cambio en la signatura de la métrica.

Si  $r < \frac{GM}{4c^2}$  entonces ocurre que la coordenada temporal se convierte en coordenada espacial y la coordenada espacial  $r$  se convierte en temporal dependiendo de esta última los coeficientes de la métrica, es decir no es una métrica estática.[4]

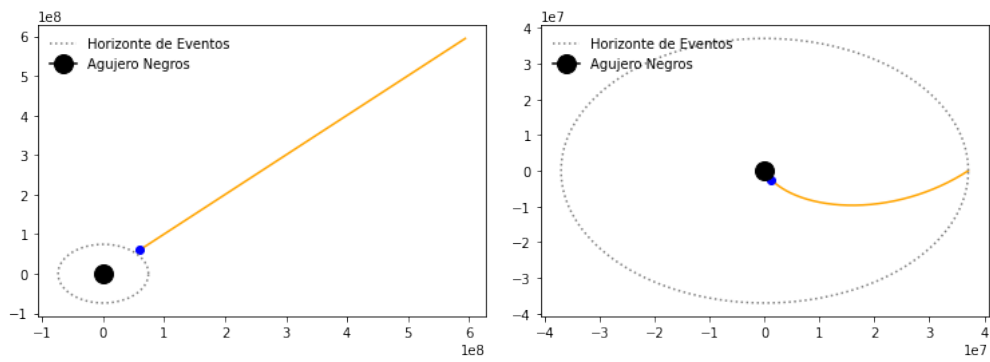


Figura 4.22: Ejemplo de geodésicas con la métrica de Schwarzschild en coordenadas cartesianas isotrópicas con un agujero negro de  $8,2 \times 10^{34} Kg$

## Capítulo 5

# Conclusiones

El objetivo de este trabajo ha sido la introducción del concepto de variedad diferenciable y dotarla de un métrica para poder estudiar la conexión de Levi-Civita, la ecuación de las curvas geodésicas y las componentes del tensor curvatura. Para ello se han escrito una serie de funciones en Python:

- Cálculo de los símbolos de Christoffel usando la ecuación (3.47).
- Cálculo de las curvas geodésicas y su resolución mediante métodos numéricos.
- Cálculo de las componentes del tensor de Riemann a partir de la ecuación (3.56).
- Cálculo de las componente del tensor curvatura a partir de la ecuación (3.65).
- Cálculo de las componentes del tensor de Ricci y curvatura escalar a partir de las ecuaciones (3.72) y (3.73) respectivamente.

El estudio de la conexión de Levi-Civita, las curvas geodésicas y las componentes del tensor curvatura se ha realizado con tres métricas: métrica de Poincaré, métrica de Minkowski y la métrica de Schwarzschild. En estas dos últimas métricas se han estudiado más a fondo con el objetivo de ser una introducción geométrica a la Relatividad General. Además, se han demostrado algunos resultados clásicos sobre los agujeros negros.

Para la solución de las geodésicas mediante métodos numéricos se ha utilizado el método de Runge-Kutta de orden 4 de paso fijo. Posteriormente se ha modificado la implementación del método de Runge-Kutta haciendo que el paso variase en función de la variación del módulo de la velocidad, mejorando los resultados obtenidos.

La conclusión más importante que he obtenido de este trabajo ha sido que una buena elección de las cartas puede resolver la mayoría de los problemas. Por ejemplo, en el estudio de las geodésicas en la métrica de Minkowski al observar las ecuaciones de las curvas geodésicas en la carta esférica no se pueden resolver tan fácilmente como con las coordenadas cartesianas, que se puede observar que son líneas rectas, esto también ocurre en la métrica de Schwarzschild, en las coordenadas esféricas es más sencillo de encontrar las soluciones que en las coordenadas cartesianas. A su vez, en los métodos numéricos también se puede apreciar que una buena selección de las cartas puede ayudar a la hora de resolverlo. En el caso de la métrica de Schwarzschild con las coordenadas esféricas el paso temporal tenía que ser de  $10^{-5}s$  para hacer que el módulo se mantuviera constante, como se demostró en la proposición 3.6.11, mientras que si se tomaba la métrica de Schwarzschild en coordenadas isotrópicas cartesianas se podía iterar con pasos temporales muy grandes y el módulo era constante.

Un futuro trabajo podría ser realizar un estudio de las geodésicas en otras métricas como la previamente mencionada *métrica de Kruskal-Szekeres* o la métrica de Alcubierre, de la cual sólo se han estudiado sus símbolos de Christoffel en este trabajo. Otro posible trabajo sería una mejora de la implementación que se ha hecho del método de Runge-Kutta y la automatización de todo el proceso de cálculo de las geodésicas.

# Bibliografía

- [1] M. Alcubierre. The warp drive: hyper-fast travel within general relativity. *Class. Quantum Grav.*, 11:L73–L77, 1994. doi:10.1088/0264-9381/11/5/001
- [2] W. M. Boothby *An introduction to differentiable manifolds and riemannian geometry*
- [3] Fernando Chamizo, Apuntes del curso de Geometría y topología impartido durante el curso 2016/2017 en la Universidad Autónoma de Madrid.  
<https://github.com/apuntes-uam-infomat/apuntes/blob/master/Geometria%20y%20Topologia/GeoTopo17.pdf>
- [4] Fernando Chamizo, Seminario 2001: una odisea en el espacio-tiempo. 2001. <http://matematicas.uam.es/~fernando.chamizo/libreria/fich/APseminario02.pdf>
- [5] do Carmo, M.P. *Riemannian Geometry* 1992 Birkhäuser.
- [6] Dongsheng Yang. *Build Prognostic Nomograms for Risk Assessment Using SAS*.
- [7] K. Gödel. An Example of a New Type of Cosmological Solutions of Einstein's Field Equations of Gravitation. *Rev. Mod. Phys.*, 21:447–450, 1949. doi:10.1103/RevModPhys.21.447.
- [8] L. Gondinho y J. Natario *An Introduction to Riemannian Geometry With Applications to Mechanics and Relativity*
- [9] Toni Gil Trilles, Apuntes de la asignatura de Álgebra Lineal impartida durante el curso 2017-2018.
- [10] Página web de la empresa PeRTICA Análisis Estadísticos S.L.. <https://pertica.info/> [Accedido 7 de Julio de 2021]
- [11] Petersen, P. (2016). *Riemannian geometry*. Springer.
- [12] B. O'Neil "Semi-Riemannian Geometry With Applications to Relativity" Academic Press, 1983
- [13] A. Mishchenko y A. Fomenko *A course of Differential Geometry and Topology*. 1988 Mir.

- [14] The WE-Heraeus International Winter School on Gravity and Light. (2015, February 10). Lecture 5: Tangent Spaces (International Winter School on Gravity and Light 2015) [Video]. YouTube. [https://www.youtube.com/watch?v=pepU\\_7NJSGM](https://www.youtube.com/watch?v=pepU_7NJSGM)
- [15] Thomas Mueller, Frank Grave, Catalogue of Spacetimes, 2010 <https://arxiv.org/abs/0904.4184v3>
- [16] C.W. Misner, K.S. Thorne, and J.A. Wheeler. Gravitation. W. H. Freeman, 1973.



## Anexo A

# Código Nomograma

---

```
1 import pandas as pd
2 import numpy as np
3
4 from IPython.display import HTML
5
6
7 import saspy as sp
8 import os
9 os.environ["PATH"] += ";C:\\Program Files\\SASHome\\SASFoundation\\9.4\\core\\sasext"
10
11 def sacar_tablas():
12
13
14     es_correcto = False
15     print("Los números se deben de introducir en formato americano (X.Y)")
16     while not es_correcto:
17         numero_variables = int(input("Introduce el número de Variables:\t"))
18         es_correcto = True if input(";Los datos son correctos?(S/N) ").upper() == 'S'
19     else False
20
21     i = 0
22     #Este diccionario en este paso va a almacenar la variable junto con
23 # si es categórica o no, y luego cada nivel, con
24 # su valor del beta y el absolute beta value
25 # se identificara con 0 una variable categórica
```

```

26     #Se indentificara con 1 una variable continua
27     diccionario_variable = {}
28     diccionario_max_min = {}
29
30     variables = []
31     betas = []
32     v_var_global = []
33     es_correcto = False
34     while not es_correcto:
35         intercept = float(input("Introduce el estimador intercept:\t"))
36         es_correcto = True if input("¿Los datos son correctos?(S/N) ").upper() == 'S' else
37
38     betas.append(intercept)
39     while i < numero_variables:
40         #Para no tener que meter todo otra vez
41         correcto = False
42         while not correcto:
43             nombre_variable = input("Introduce el nombre la variable: \t")
44             nombre_variable = nombre_variable.replace(" ", "_")
45             es_categorica = -1
46             while es_categorica not in [0,1]:
47                 try:
48                     es_categorica = int(input("¿La variable es categórica?\n\t0 si es
49 categorica \n\t1 si es continua\n"))
50                 except:
51                     print("El formato no es válido vuelva a introducirlo otra vez
52 (Debe ser un número en formato americano X.Y)")
53                     if es_categorica not in [0,1]:
54                         print("Clase de la variable introducida no es correcto.
55 Por favor vuelva a introducirlo")
56
57             vector_info_variable = [es_categorica]
58             vector_beta_variable = []
59             vector_variables_variable = []
60             max_min_var = []
61
62
63
64     #Si es categorica
65     if es_categorica == 0:
66         numero_niveles = int(input("Introduce el número de niveles:\t"))
67         j = 0

```

```

68     while j < numero_niveles:
69         nombre_nivel = input("Introduce el nombre del nivel:\t")
70         try:
71             estimador_beta = float(input("Introduce el estimador para dicho parame
72         except:
73             print("El formato no es válido vuelva a introducirlo otra vez")
74             estimador_beta = float(input("Introduce el estimador para dicho parame
75
76         vector_variables_variable.append(nombre_nivel)
77         vector_beta_variable.append(estimador_beta)
78         vector_info_variable+= [nombre_nivel, estimador_beta, estimador_beta]
79
80         j+= 1
81     #Si es continua
82     else:
83         try:
84             estimador_beta = float(input("Introduce el estimador para dicho paramet
85         except:
86             print("El formato no es válido vuelva a introducirlo otra vez")
87             estimador_beta = float(input("Introduce el estimador para dicho paramet
88         try:
89             minimo =float(input("Introduce el minimo de dicha variable:\t"))
90         except:
91             print("El formato no es válido vuelva a introducirlo otra vez")
92             minimo =float(input("Introduce el minimo de dicha variable:\t"))
93         try:
94             maximo = float(input("Introduce el maximo de dicha variable:\t"))
95         except:
96             print("El formato no es válido vuelva a introducirlo otra vez")
97             maximo = float(input("Introduce el maximo de dicha variable:\t"))
98
99         max_min_var = [maximo, minimo]
100
101         vector_info_variable += [ estimador_beta, estimador_beta*(maximo-minimo),
102     sestimador_beta*(maximo-minimo)]
103         #vector_info_variable += [ estimador_beta, estimador_beta*(maximo),estimador
104         vector_beta_variable.append(estimador_beta)
105         vector_variables_variable.append(nombre_variable)
106
107     correcto = True if input("¿Los datos son correctos?(S/N)").upper() == 'S' else
108     if correcto:
109         break

```

```

110         else:
111             print("Introduce otra vez los datos")
112             betas += vector_beta_variable
113             variables += vector_variables_variable
114             diccionario_variable[nombre_variable] = vector_info_variable
115             v_var_global.append(nombre_variable)
116             if es_categorica == 1:
117                 diccionario_max_min[nombre_variable] = max_min_var
118                 i+=1
119             print("")
120 dict_variables_unidas = diccionario_variable
121
122 #Si todos son negativos lo invertimos
123 betas_todos_neg= [1 if i < 0 else 0 for i in betas]
124
125 todos_neg = False
126 if 1 in betas_todos_neg[1:] and not 0 in betas_todos_neg[1:] and len(betas) > 2:
127     todos_neg = True
128     copia_betas = [i for i in betas]
129     betas =[betas[0]] + [abs(betas[i]) for i in range(1, len(betas))]
130
131
132
133
134 asociacion = [(variables[i],betas[i +1]) for i in range(len(variables))]
135
136 #Diccionario_variables es un diccionario que
137 # empara cada variable de la regresión con su beta
138 #Lo que hago con (len(betas)-len(variables)
139 #es para la diferencia de quitar todos los intercept que aparecen. s
140     diccionario_variables = {asociacion[i][0]:
141 [asociacion[i][1]] for i in range(0,len(asociacion))}
142
143
144 #Ordeno las variables según su absolute beta value maximo
145 v = []
146 #Aquí hay que poner el abs() si no funciona
147
148 for var in v_var_global:
149     v.append((var,max([i for i in dict_variables_unidas[var][3::3])))
150
151     #Prueba con el abu que no sea absolute porque puede ocurrir que los betas sean ne

```

```

152     #Como dice el artículo:
153     v_var_global.sort(key= lambda var: max(
154 [i for i in dict_variables_unidas[var][3::3]]), reverse = True)
155     #v_var_global.sort(key= lambda var: max(
156 [i for i in dict_variables_unidas[var][3::3]]), reverse = True)
157
158
159     #Seleccionamos la que mayor ABV tiene:
160     max_var = v_var_global[0]
161     tipo_max_var = dict_variables_unidas[max_var][0]
162
163     #Saco los Maximos LP y los mínimos LP
164     #Si es continua
165     if tipo_max_var == 1:
166         #Saco el beta asociado a esa variable
167         beta_continua = dict_variables_unidas[max_var][1]
168
169
170         #Calculo sus maximos predictores lineales
171         max_LP = beta_continua*diccionario_max_min[max_var][0]
172         min_LP = beta_continua*diccionario_max_min[max_var][1]
173
174     #Si es categorica siempre va a estar como 0 o 1
175     else:
176
177         #Calculo los LP
178
179         vector_beta = [v for v in dict_variables_unidas[max_var][2::3]]
180
181         max_LP = max(vector_beta)* 1
182
183         #Si es categórica siempre va a ser 0 porque es el mínimo
184         min_LP = 0
185
186     puntos_por_unidad_de_lp = 100/(max_LP-min_LP)
187
188     print(max_LP, min_LP)
189
190     suma_min_abs = abs(betas[0])
191     suma_min =betas[0]
192     for variable in v_var_global:
193         #Si es continua

```

```

194     print(variable,dict_variabes_unidas[variable][0] )
195     if dict_variabes_unidas[variable][0] == 1:
196
197         suma_min += dict_variabes_unidas[variable][1]
198 *diccionario_max_min[variable][1]
199
200         suma_min_abs += abs(dict_variabes_unidas[variable][1])
201 *diccionario_max_min[variable][1]
202     else:
203
204         vector_beta = [v for v in dict_variabes_unidas[variable][2::3]]
205         #Si solo es de 0, 1
206         if len(vector_beta) == 1:
207             suma_min += 0
208         else:
209             suma_min += min(vector_beta)
210             suma_min_abs += abs(min(vector_beta))
211
212     suma_max = betas[0]
213     suma_max_abs= abs(betas[0])
214     for variable in v_var_global:
215         #Si es continua
216         if dict_variabes_unidas[variable][0] == 1:
217             suma_max += dict_variabes_unidas[variable][1]
218 *diccionario_max_min[variable][0]
219             suma_max_abs +=abs(dict_variabes_unidas[variable][1])
220 * diccionario_max_min[variable][0])
221         #Si es categórica
222         else:
223             vector_beta = [v for v in dict_variabes_unidas[variable][2::3]]
224             suma_max += max(vector_beta)
225             suma_max_abs += abs(max(vector_beta))
226
227     escritor_salida = open("nomograma.txt","w")
228     escritor_salida.write("y          y_label          x_label          x          low
229 high          x_label2\n")
230
231     #Saco los totalpoints
232     #
233     if tipo_max_var == 1 :
234         min_total_points = puntos_por_unidad_de_lp * (min_LP-min_LP)
235         max_total_points =puntos_por_unidad_de_lp *(max_LP-min_LP)

```

```

236     else:
237         min_total_points = (puntos_por_unidad_de_lp *
238 (suma_min - (betas[0])))
239         max_total_points =(puntos_por_unidad_de_lp *
240 (suma_max - (betas[0])))#(puntos_por_unidad_de_lp *
241 (suma_max_abs- abs(betas[0])))
242
243
244     min_risk_of_y = np.e**(suma_min)/(1+np.e**(suma_min))
245     max_risk_of_y = np.e**(suma_max)/(1+np.e**(suma_max))
246
247     inicio_bucle = min_risk_of_y
248     fin_bucle =max_risk_of_y
249
250
251     #Primero escribo las tablas risk of even total points
252     cadena_risk_of_event = ""
253     #Para que comience en el siguiente
254     partes = 7
255     pos = 1
256
257     if not todos_neg:
258         i = ( inicio_bucle+betas[0])/puntos_por_unidad_de_lp
259         cadena_risk_of_event = "{}\t {}\t {:.2f}\t {:.2f}\t
260 {}\t {}\t {}\n".format(1,"Risk of Event",
261 min_risk_of_y,0, 0.9,1, "")
262
263         cadena_total_points = "{}\t {}\t {:.2f}\t {:.2f}\t
264 {}\t {}\t {}\n".format(2,"Total Points",
265 round(min_total_points),0, 1.9,2, "")
266
267         partes = 7
268         pos = 1
269
270         lp= min_total_points + (max_total_points -
271 min_total_points)/partes
272
273         while lp < abs(max_total_points):
274             puntos = (lp)/puntos_por_unidad_de_lp+betas[0]
275 +len(v_var_global)*min_LP#len(diccionario_max_min.keys())*min_LP
276             risk_of_y = np.e**puntos/(1+np.e**puntos)
277             total_point = lp

```

```

278         x_label2 = ""
279         #           y y_label x_label x low
280 high x_label
281         cadena_risk_of_event += "{}\t {}\t {:.2f}\t {:.2f}\t {}\t {}\t {}\n".format(1
282 "Risk of Event", round(risk_of_y,2),pos/(partes+1)*100, 0.9,1, x_label2)
283
284         cadena_total_points += "{}\t {}\t {:.2f}\t {:.2f}\t {}\t {}\t {}\n".format(2,
285 "Total Points", round(total_point),pos/(partes+1)*100, 1.9,2, "")
286
287         lp += (max_total_points - min_total_points)/(partes+1)#(abs(max_total_points) -
288 abs(min_total_points)).7
289         pos += 1
290
291         puntos = (lp)/puntos_por_unidad_de_lp+betas[0]
292         risk_of_y = np.e**puntos/(1+np.e**puntos)
293
294         cadena_risk_of_event += "{}\t {}\t {:.2f}\t {:.2f}\t {}\t {}\t {}\n".format(1
295 "Risk of Event", max_risk_of_y,100, 0.9,1, "")
296
297         cadena_total_points+= "{}\t {}\t {:.2f}\t {:.2f}\t {}\t {}\t {}\n".format(2,
298 "Total Points",round(max_total_points) ,100, 1.9,2, "")
299
300         #Si todos los betas son negativos le damos la vuelta a la cadena de risk of event
301         else:
302             i = ( inicio_bucle+betas[0])/puntos_por_unidad_de_lp
303             cadena_risk_of_event = "{}\t {}\t {:.2f}\t {:.2f}\t {}\t {}\t {}\n".format(1,
304 "Risk of Event",max_risk_of_y,0, 0.9,1, "")
305
306             cadena_total_points = "{}\t {}\t {:.2f}\t {:.2f}\t {}\t {}\t {}\n".format(2,
307 "Total Points", round(min_total_points),0, 1.9,2, "")
308             partes = 7
309             pos = 1
310             #bulce para total points
311             incremento = (max_total_points - min_total_points)/partes
312             lp = min_total_points + incremento
313             while lp < max_total_points:
314                 total_point = lp
315                 x_label2 = ""
316                 cadena_total_points += "{}\t {}\t {:.2f}\t {:.2f}\t {}\t {}\t {}\n".format(2,
317 "Total Points", round(total_point),pos/(partes+1)*100, 1.9,2, "")
318
319                 pos += 1

```



320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361

```
lp += incremento

#bucle del risk of y
pos = 1
incremento = (max_risk_of_y - min_risk_of_y)/partes
riesgo = max_risk_of_y - incremento

while riesgo > (min_risk_of_y+incremento):
    x_label2 = ""
    cadena_risk_of_event += "{}\t {}\t {:.2f}\t {:.2f}\t {}\t {}\t {}\n".format(1,
    "Risk of Event",
    round(riesgo,2),pos/(partes+1)*100, 0.9,1, x_label2)
    riesgo -= incremento
    pos += 1

    cadena_risk_of_event += "{}\t {}\t {:.2f}\t {:.2f}\t {}\t {}\t {}\n".format(1
    ,"Risk of Event",
    round(min_risk_of_y,2),pos/(partes+1)*100, 0.9,1, x_label2)

    cadena_total_points += "{}\t {}\t {:.2f}\t {:.2f}\t {}\t {}\t {}\n".format(2,
    "Total Points", round(max_total_points),pos/(partes+1)*100, 1.9,2, "")

    escritor_salida.write(cadena_risk_of_event)
    escritor_salida.write(cadena_total_points)

y = 3
for variable in v_var_global:

    datos_var = dict_variables_unidas[variable]

    tipo = datos_var[0]

    #Si la variable es numerica
    if tipo == 1:
        beta = datos_var[1]
        print(beta)
        maxi =diccionario_max_min[variable][0]
        mini =diccionario_max_min[variable][1]
```

```

362
363     variable_contenido = mini
364     incremento = (maxi-mini)/10
365
366     for n in range(0, 11):
367         n= (n+1)*10
368
369         lp = variable_contenido*beta
370
371         x = round(100*(lp-min_LP)/(max_LP-min_LP))
372         y_label = variable
373         x_label = variable_contenido
374         low = y -0.1
375         high = y
376         x_label2 = ""
377         if n == 100 and beta < 0 and diccionario_max_min[variable][0]>0 and
378 diccionario_max_min[variable][1]>0:
379             x_label2 = "Substract"
380             variable_contenido += incremento
381             #         y y_label x_label x low high x_label
382
383             cadena = "{}\t {}\t {:.2f}\t {:.2f}\t {}\t {}\t {}\n".format(y, y_label,
384 x_label, abs(x), low, high, x_label2)
385             escritor_salida.write(cadena)
386         # Si es categorica
387         if tipo == 0:
388             vector_info = datos_var[1:]
389             nombres_categorias = vector_info[0::3]
390             betas = vector_info[1::3]
391             categoria = 0
392             #Si solo tiene dos niveles
393             if len(nombres_categorias) == 2:
394                 beta = betas[0]
395                 low = y -0.1
396                 high = y
397                 y_label = variable
398                 nombre_etiqueta = dict_variables_unidas[variable][1::3]
399                 #Con la puntuación 0
400                 x = round(100*(beta-min_LP)/(max_LP-min_LP))
401                 x_label2 =nombre_etiqueta[0]
402                 x_label = ""
403                 cadena = "{}\t {}\t {}\t {:.2f}\t {}\t {}\t {}\n".format(y, y_label,

```

```

404 x_label.replace(" ", ""),abs(x), low, high, x_label2.replace(" ", ""))
405 escritor_salida.write(cadena)
406
407 #Con la puntuacion 1
408 beta =betas[1]
409 x = round(100*(beta-min_LP)/(max_LP-min_LP))
410 if beta < 0:
411     x_label2 = "Substract"
412 else:
413     x_label2 = "Add"
414
415 x_label = nombre_etiqueta[1]
416 cadena = "{}\t {}\t {}\t {:.2f}\t {}\t {}\t {}\n".format(y, y_label,
417 x_label.replace(" ", ""),abs(x), low, high, x_label2.replace(" ", ""))
418 escritor_salida.write(cadena)
419
420 else: #Si tiene más de dos niveles
421 nombre_etiqueta = dict_variables_unidas[variable][1::3]
422 nombre = nombre_etiqueta[0]
423 #Para la variable de referencia
424 escritor_salida.write("{}\t {}\t {}\t {:.2f}\t {}\t {}\t {}\n".format
425 (y,variable, nombre.replace(" ", ""),0, y-0.1, y, ""))
426 while (categoria) < (len(vector_info)//3):
427     #Nombre de la categoria
428     cat =nombre_etiqueta[categoria]
429     beta = betas[categoria]
430
431     x =round(100*(beta-min_LP)/(max_LP-min_LP))
432
433     x_label = cat
434     low = y -0.1
435     high = y
436     y_label = variable
437     x_label2 = ""
438     x_label = cat
439     if beta <0 :
440         x_label2 = "Substract"
441     else:
442         x_label2 = "Add"
443
444     if x == 0:
445         x_label2 =cat

```

```

446         x_label = ""
447         if beta < 0:
448             x_label = "Substract"
449         else:
450             x_label = "Add"
451         cadena = "{}\t {}\t {}\t {:.2f}\t {}\t {}\t {}\n".format(y, y_label,
452 x_label.replace(" ", ""),abs(x), low, high, x_label2.replace(" ", ""))
453         escritor_salida.write(cadena)
454         categoria += 1
455     y+= 1
456     #Ahora rellenamos la variable de control
457
458     categoria = "Points"
459     for n in range(0, 11):
460         n = 10*n
461         y_label = categoria
462         x_label = n
463         x= n
464         low = y-0.1
465         high = y
466         x_label2 = ""
467         cadena = "{}\t {}\t {}\t {}\t {}\t {}\t {}\n".format(y, y_label, x_label, x, low,
468 high, x_label2)
469         escritor_salida.write(cadena)
470     escritor_salida.close()
471     return v_var_global
472
473
474 def leer_datos_completos(nombre_fichero="datos.txt"):
475     try:
476         fichero = open(nombre_fichero, "r")
477     except:
478         print("No existe el fichero", nombre_fichero)
479         fichero = input("Por favor introduzca el nombre del fichero que contiene los datos: ")
480     #Leemos la cabecera
481     cabecera = fichero.readline()
482     cabecera = cabecera[:-1]
483     cabecera.replace(" ", " ")
484     fichero.close()
485
486     data = pd.read_csv(nombre_fichero, sep=" ")
487     data.columns= [uwu.upper() for uwu in cabecera.split(" ")]

```

```

488
489
490     return data
491
492 def crear_nomograma(variables_orden, dir):
493     nomograma = leer_datos_completos("nomograma.txt")
494
495     proc_format = "proc format; \n value yfmt \n"
496     fin = "2 = 'TOTAL POINTS' \n 1 = 'RISK' \n 0 = ' ';"
497     for i in range(len(variables_orden), -1, -1):
498         if i == len(variables_orden):
499             proc_format += "{} = '{}'\n".format(i+3, "POINTS" )
500         else:
501             proc_format += "{} = '{}'\n".format(i+3, variables_orden[i] )
502     proc_format += fin
503
504
505     leer = """ data WORK.kkk      ;
506     %let _EFIERR_ = 0; /* set the ERROR detection macro variable */
507     infile 'nomograma.txt' delimiter='09'x MISSOVER DSD lrecl=32767 firstobs=2 ;
508         informat y best32. ;
509         informat y_label $15. ;
510         informat x_label $10. ;
511         informat x best32. ;
512         informat low best32. ;
513         informat high best32. ;
514         informat x_label2 $10. ;
515         format y best12. ;
516         format y_label $15. ;
517         format x_label $10. ;
518         format x best12. ;
519         format low best12. ;
520         format high best12. ;
521         format x_label2 $10. ;
522     input
523         y
524         y_label $
525         x_label $
526         x
527         low
528         high
529         x_label2 $

```

```

530     ;
531     if _ERROR_ then call symputx('_EFIERR_',1); /* set ERROR detection macro variable */
532     run;\n
533     """
534     proc sgplot = r"ods graphics on/ imagefmt=svg imagename='SVG';" + "\n" + r" ODS LISTING O
535 'C:\Users\fgonzalez\Desktop\mis_proyectos\nomograma\UntitledFolder\libreria_python' ;"+ "\n
536 + " proc sgplot data
537 = kkk noautolegend ;"+ "\n" + " series x = x y = y/group = y_label lineattrs = (pattern =
538 1 thickness = 2);"+ "\n"+ "
539 highlow x=x high = high low=low / lowlabel=x_label highlabel=x_label2; yaxis display=
540 ( nolabel noline noticks )
541 tickvalueformat =yfmt. VALUES=(0 TO {} BY 1);".format(len(variables_orden)+3)+"\n"+ "
542 xaxis display= none ; run;"
543     subir = leer + proc_format + proc_sgplot
544
545
546     escritor_fichero_sas = open(r"{}\Nomograma.sas".format(dir), "w")
547     escritor_fichero_sas.write(subir)
548     escritor_fichero_sas.close()
549     try:
550         sas =sp.SASsession(cfgname='winlocal', results='Pandas', encoding='utf-8')
551         codigo_sas = open(r"{}\Nomograma.sas".format(dir), "r")
552         codigo = codigo_sas.read()
553
554         return_sas = sas.submit(codigo)
555         codigo_sas.close()
556     except:
557         print("Debes de instalar la librería SASPy, pero se ha creado un archivo .sas que
558 contiene el código para ejecutarlo.")
559         return False
560
561     return True
562
563
564 variables_orden= sacar_tablas()
565
566 dir = os.getcwd()
567 v = crear_nomograma(variables_orden, dir)
568 print(v)

```

---

## Anexo B

# Estructuras maximales

En este apéndice se demostrará que toda estructura diferenciable que no sea maximal existe una estructura diferenciable maximal a la que pertenece.

**Proposición B.0.1** (Ver [12]). *Sea  $\mathcal{M}$  un espacio Hausdorff y 2AN. Sea  $\mathcal{V} = \{(V_\beta, \phi_\beta)\}_{\beta \in I}$  un recubrimiento de  $\mathcal{M}$  y un entorno coordinado  $\mathcal{C}^\infty$ -compatibles. Entonces existe una única estructura de clase  $\mathcal{C}^\infty$  diferenciable en  $\mathcal{M}$  que contiene a este entorno coordinado.*

*Demostración.* Para probar que existe una variedad que contiene a  $\mathcal{V}$  se va a construir una estructura diferenciable compatible con  $\mathcal{M}$ .  $\mathcal{U} = \{(U_i, \psi_i)\}_{i \in I'}$  de modo que son una familia de todos los entornos coordinados de  $\mathcal{M}$  de  $\mathcal{C}^\infty$ -compatibles con cada uno de los  $(V_\beta, \psi_\beta)$ .

Una vez definido  $\mathcal{U}$  se va a demostrar, siguiendo la definición 3.3.2, que es una estructura diferenciable de  $\mathcal{M}$ .

1.  $\bigcup_{i \in I'} U_i = \mathcal{M}$  debido a que  $\mathcal{U}$  contiene a  $\mathcal{V}$  y este recubría a  $\mathcal{M}$ .
2. Ahora vamos a ver que cumple la segunda propiedad. Para ello vamos a tomar dos entornos coordinados  $(U_i, \psi_i)$  y  $(U'_i, \psi'_i)$  que pertenece a  $\mathcal{U}$  y  $U' \cap U \neq \emptyset$ , entonces tenemos que ver que  $\forall \alpha, \beta \in I'$ :
  - $\psi_i(U_i) \cap \psi'_i(U'_i) = W \neq \emptyset$  Tal y como hemos definido nuestra familia de entornos coordinados tenemos que son cambios de coordenadas y homeomorfismos bien definidos de abiertos de  $\mathbb{R}^n$
  - Como hemos dicho en el punto anterior tenemos que son cambios de coordenadas y por lo tanto  $\psi_i^{-1}(U_i)$  y  $\psi'_i^{-1}(U'_i)$  son abiertos de  $\mathbb{R}^n$

- Ahora nos falta demostrar que  $\psi_i \circ \psi'_i$  es diferenciable. Sea  $x = \psi(x) \in \psi(U \cap U')$  entonces tenemos que  $p \in V_\beta$  para uno de los entornos coordenados de  $\mathcal{V}$ . Por lo tanto hay un abierto  $W$  tal que  $p \in W = V_\beta \cap U \cap U'$  y  $x \in \psi(W)$ , que también es abierto. Entonces tenemos que  $\psi_i \circ \psi'_i = \psi_i \circ \phi_\beta^{-1} \circ \phi_\beta \circ \psi'_i$  en  $\psi(W)$  pero como  $(U_i, \psi_i)$  y  $(U'_i, \psi'_i)$  son  $\mathcal{C}^\infty$ -compatibles con  $(V_\beta, \phi_\beta)$  entonces  $\psi_i \circ \phi_\beta^{-1}$  y  $\phi_\beta \circ \psi'_i$  son diferenciables y por ende  $\psi_i \circ \psi'_i$  también lo es en  $\psi(W)$ . Y en consecuencia se obtiene que es diferenciable en todo entorno coordenado de su dominio.
3. Esta propiedad es directa debido a que para cualquier  $(U, \psi)$  que sea compatible con todos los entornos coordenados de la familia que hemos construido; en concreto lo es respecto a la familia inicial  $\mathcal{V}$ .

Y por lo tanto  $\mathcal{U} = \{(U_i, \psi_i)\}_{i \in I'}$  es una estructura diferenciable maximal.

□



## Anexo C

# Runge Kutta Orden 4

---

```
1  # metodo de Runge-Kutta de orden 4 para un sistema de N eqs
2
3
4  import os
5
6
7  #La función como tal
8  def f(j, x):
9      #j nos indica si queremos el 0 o el 1
10     #x[0]=x y x[1]=y
11     resultado = 0
12
13
14     #En el caso de que haya más de 2 dimensiones añadir un elif
15     if j == 0:
16         try:
17
18             resultado += x[2]
19         except:
20             print("Overflow error")
21             return
22     elif j == 1:
23         try:
24
25             resultado += x[3]
```

```

26         except:
27             print("Overflow error")
28             return
29     elif j == 2:
30         try:
31             resultado = 2*(x[2]*x[3])/x[1]
32         except:
33             print("Overflow error")
34             return
35     else:
36         try:
37
38             resultado = (-x[2]**2+x[3]**2)/x[1]
39         except:
40             print("Overflow error")
41             return
42     return resultado
43
44
45 def F(i, j , x, w, dim, h):
46     #dim es la dimensión
47     # h es el paso
48     r = [0 for a in range(dim)]
49     v = 0
50
51     if i == 0:
52         for a in range(0, dim):
53             r[a] = x[a]
54
55     elif i == 1 or i == 2:
56         for a in range(dim):
57             r[a] = x[a]+ (h/2)* w[i-1][a]
58     else: #si es 3
59         for a in range(dim):
60             r[a] = x[a] + h*w[i-1][a]
61
62     v = f(j, r)
63     return v
64
65
66
67 def main():

```

```

68
69 N = 4 # el tamaño del espacio de estados
70
71 y= [0 for i in range(N)]
72
73 w = [[0 for j in range(N)] for i in range(0,4)]
74
75 k = 0
76
77 print("Introducir tiempo máximo:")
78 t= 0
79 #tmax = int(input())
80 tmax=100
81
82 #Leo los datos iniciales
83 for j in range(0, N):
84     print("Introduce x[{}]:\t".format(j))
85     y[j] = float(input())
86
87 #Introducir el paso temporal un buen h = 0.001
88 print("Introduce el paso temporal: \t")
89 h = float(input()) #Puede ser positivo o negativo
90 #h= 0.001
91 #creo y abro el fichero de salida
92 nombre_fichero = "datospython{}-{}.dat".format(y[0], y[1])
93 fichero_salida = open(nombre_fichero, 'w')
94 fichero_salida.write("# metodo de R-K segun rk.c \n")
95
96 #Escribo el punto inicial
97 esc = "{:20.20f}".format(t)
98 fichero_salida.write(str(esc))
99
100 for d in y:
101     fichero_salida.write("\t {:20.20f}".format(d))
102 fichero_salida.write("\n")
103 #Iteraciones de Runge-Kutta
104 while abs(t) < tmax:
105
106     k += 1
107     #Aumento el tiempo
108     t += h
109

```

```

110
111     #coeficientes del método RK
112     for i in range(0, 4):
113         for j in range(0, N):
114             w[i][j] = F(i, j, y, w, N, h)
115             if w[i][j] is None:
116                 fichero_salida.close()
117                 return
118
119     #escribo el tiempo
120     esc = "{:20.20f}".format(t)
121     fichero_salida.write(str(esc))
122
123     #Solucion numerica
124     for j in range(0, N):
125         y[j] += (w[0][j] + 2*(w[1][j] + w[2][j]) + w[3][j]) * h/6
126         res_aux = "\t {:20.20f}".format(y[j])
127         fichero_salida.write(res_aux)
128         fichero_salida.write("\n")
129         if y[1] < 10**-5:
130             break
131     #Cierro el fichero
132     fichero_salida.close()
133
134     return
135
136
137 if __name__ == "__main__":
138     main()

```

---

La implementación de Runge-Kutta con el paso variable para mantener el módulo, según la proposición 3.6.11.

---

```

1 import os
2 from math import sin, cos, sqrt, pi, floor
3 import numpy as np
4
5 #La función como tal
6 def f(j, y):

```

```

7      #j nos indica si queremos el 0 o el 1
8      #y[0]=t ; y[1]=x; y[2]=y
9
10     resultado = 0
11     #Masa del objeto que deforma el cuerpo
12
13     M=2*10**35
14     #Agujero centro via lactea
15     #M = 8.2*10**36
16     #Constante de gravitación universal
17     G=6.674*10**(-11) # N*m**2/kg**2
18     #N =m/s**2
19     #Velocidad de la luz
20     c = 3*10**8#m/s**
21
22     #Defino las variables para simplificar
23     rho_s = G*M/(2*c**2)
24
25     rho= sqrt(y[1]**2+y[2]**2)
26
27     if j ==0:
28         try:
29
30             resultado = +y[3]
31
32         except:
33             print("Overflow error")
34             return
35     elif j ==1:
36         try:
37
38             resultado =y[4]
39         except:
40             print("Overflow error")
41             return
42     elif j ==2:
43         try:
44
45             resultado =y[5]
46         except:
47             print("Overflow error")
48             return

```

```

49     elif j ==3:
50         try:
51             resultado = -4*rho_s*y[3]/(rho*(rho**2-rho_s**2))*(y[4]*y[1]+y[2]*y[5])
52         except:
53
54             print("Overflow error")
55             return
56     elif j ==4:
57         try:
58             resultado = -2*c**2*(rho-rho_s)*rho**(3/2)*rho_s*y[1]*y[3]**2/(rho_s+rho)**7-2*
59         except:
60
61             print("Overflow error")
62             return
63     else:
64         try:
65             resultado = -2*c**2*(rho-rho_s)*rho**(1.5)*rho_s*y[2]*y[3]**2/(rho_s+rho)**7-2*
66         except:
67
68             print("Overflow error")
69             return
70
71     return resultado
72
73
74 def F(i, j , x, w, dim, h):
75     #dim es la dimensión
76     # h es el paso
77     r = [0 for a in range(dim)]
78     v = 0
79
80     if i == 0:
81         for a in range(0, dim):
82             r[a] = x[a]
83
84     elif i == 1 or i == 2:
85         for a in range(dim):
86             r[a] = x[a]+ (h/2)* w[i-1][a]
87     else: #si es 3
88         for a in range(dim):
89             r[a] = x[a] + h*w[i-1][a]
90

```

```

91     v = f(j, r)
92     return v
93
94
95
96 def main():
97
98     N = 6 # el tamaño del espacio de estados
99
100    y= [0 for i in range(N)]
101
102    w = [[0 for j in range(N)] for i in range(0,4)]
103
104    k = 0
105
106    #print("Introducir tiempo máximo:")
107    t= 0
108    #tmax = int(input())
109    tmax=5*10**20
110
111    #Leo los datos iniciales
112    nombres_variiables=["Coordenada temporal", "Coordenada X", "Coordenada Y" , "Aceleración
113    #for j in range(0, N):
114    #    print("Introduce {}:\t".format(nombres_variiables[j]))
115    #    y[j] = float(input())
116
117
118    #Cambiamos las variables
119    # t_0
120    y[0]=0
121    #x_0
122    y[1]=3.5*10**9+1#/(2**9)
123    #y_0
124    y[2]=3.5*10**9+1
125
126    #Las velocidades
127    #\dot{t}
128    y[3]=1
129    #\dot{x}
130    y[4]=-100
131    #\dot{y}
132    y[5]=-100

```

```

133
134
135 #Introducir el paso temporal un buen h = 0.001
136 #print("Introduce el paso temporal: \t")
137 #h = float(input()) #Puede ser positivo o negativo
138 h= 10
139 #creo y abro el fichero de salida
140 nombre_fichero = "adaptativo_iso_schw_{}-{}.dat".format(y[1], y[2])
141 fichero_salida = open(nombre_fichero, 'w')
142 print(nombre_fichero)
143 #Escribo el punto inicial
144 esc = "{:20.20f}".format(t)
145 fichero_salida.write(str(esc))
146
147 for d in y:
148     fichero_salida.write("\t {:.20.20f}".format(d))
149 fichero_salida.write("\n")
150 #Iteraciones de Runge-Kutta
151 every = 1000
152 e = 0
153 while abs(t) < tmax:
154
155     k += 1
156     #Aumento el tiempo
157     t += h
158
159
160 #coeficientes del método RK
161 for i in range(0, 4):
162     for j in range(0, N):
163         w[i][j] = F(i, j, y, w, N, h)
164         if w[i][j] is None:
165             fichero_salida.close()
166             return
167
168 #escribo el tiempo
169 if e == every:
170     esc = "{:20.20f}".format(t)
171     fichero_salida.write(str(esc))
172
173 #Solucion numerica
174 for j in range(0, N):

```



```

175         y[j] += (w[0][j] + 2*(w[1][j] + w[2][j]) + w[3][j]) * h/6
176
177         if e == every:
178             res_aux = "\t {:.20f}".format(y[j])
179             fichero_salida.write(res_aux)
180     if e == every:
181         fichero_salida.write("\n")
182         e = 0
183     e += 1
184
185     M = 2*10**35
186     #Agujero centro via lactea
187     #M = 8.2*10**36
188     #Constante de gravitación universal
189     G = 6.674*10**(-11) # N*m**2/kg**2
190     #N = m/s**2
191     #Velocidad de la luz
192     c = 3*10**8 #m/s*
193
194     rho_s = G*M/(2*c**2)
195     t, x, yy, dt, dx, dy = y[0], y[1], y[2], y[3], y[4], y[5]
196     z, dz = 0, 0
197     rho = sqrt(x**2 + yy**2 + z**2)
198     g = np.matrix([[ -c**2*((1-rho_s/rho)/(1+rho_s/rho))**2, 0, 0, 0],
199                   [ 0, (1+rho_s/rho)**4, 0, 0],
200                   [ 0, 0, (1+rho_s/rho)**4, 0],
201                   [ 0, 0, 0, (1+rho_s/rho)**4]])
202     v = np.array([dt, dx, dy, dz])
203     l = np.matmul(v, g)
204     l = np.matmul(l, v)
205     l = l[0]
206     if (abs(t)) <= (h+h/1000):
207         g_ant = l
208         y_ant = y
209     elif abs(t) > h and ((1-g_ant) < 0.1 and (1-g_ant) < -0.1) :
210         g_ant = l
211     else:
212         t -= h
213         h = h/2
214         every = int(every/2)
215         e = 0
216         y = y_ant

```

```
217         break
218     y_ant = y
219
220     #Cierro el fichero
221     fichero_salida.close()
222
223     return
224
225
226 if __name__ == "__main__":
227     main()
```

---