

Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de
Telecomunicación

Simulador en Python del efecto de múltiples obstáculos
sobre un radioenlace según la Rec. ITU-R P.526

Autor: María Laó Cañadas
Tutor: Susana Hornillo Mellado

Dep. Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2018



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de Telecomunicación

**Simulador en Python del efecto de múltiples
obstáculos sobre un radioenlace según la Rec. ITU-
R P.526**

Autor:
María Laó Cañadas

Tutor:
Susana Hornillo Mellado
Profesora Contratada Doctora

Dep. de Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2018

Trabajo Fin de Grado: Simulador en Python del efecto de múltiples obstáculos sobre un radioenlace según la Rec. ITU-R P.526

Autor: María Laó Cañadas
Tutor: Susana Hornillo Mellado

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2018

El Secretario del Tribunal

*A mi familia
A esa parte de mi que decía
que no podría*

Agradecimientos

El inicio del fin.

Palabras que quedarán impregnadas en este papel para siempre, por eso, las personas que aquí van a ser nombradas, también son de ese tipo, de las que estuvieron en el paso del tiempo y de las que quedaran en mi mente, retina y corazón cuando este acabe.

Las personas son lugares.

EL CORONIL, SEVILLA.

Mis padres. Este es el pilar fundamental dónde se basa toda mi vida, educación y fuerza. En este lugar he recibido todo el apoyo que he necesitado, no solo en estos años, si no en todos los anteriores en los que tuve que luchar contra mí misma para conseguir todas las metas que me proponía. Mis padres son los que le han dado forma a cada uno de mis sueños, los que me han permitido que estos años pueda estar fuera de casa, creando mi propia vida. Haciendo lo que más me gustaba.

Mi hermano. Persona que, cada vez más lejos, se hace sentir más cerca.

Mis amigas. Ellas no han podido vivir esta carrera como yo, cada una toma su camino, pero siempre queda la esencia. Cuando nos juntamos volvemos a ser nosotras, sin peros, sin preguntas, sin excusas. Solo nosotras.

CALLEJÓN FRESA, SEVILLA.

Valle, María y Laura (Posteriormente León). En casa o fuera de esta. En Sevilla o en cualquier ciudad. Con frío o con calor. Viviendo juntas o sin estarlo. Siempre habéis sido imprescindibles e irremplazables, porque, no es fácil encontrar alguien que sin ser nada, pueda acabar siendo todo.

Incontrolables, incansables, indomables, fuertes, guerreras y mujeres ingeniero, eso sois, y eso somos.

En resumen, "HOGAR".

Todas aquellas personas (que no han sido pocas) que han pasado por este lugar una o más veces. En especial, nombrar a una que conocí por sorpresa, una tarde cualquiera, Havixu. Las palabras más pequeñas son para las personas más grande. Todo te va a ir bien, te lo mereces.

CALLE SOR MILAGROS, SEVILLA

Dunai. Eres esa persona que jamás pensé que iba a conocer, que jamás pensé que podría ser mi amigo y mucho menos, que llegara a importarme tanto como tú lo haces. Desde Sevilla hasta Siracusa.

Adrián (Posteriormente Campo de los Mártires). Una vez dije algo como “quédate con quién te espera y te abraza cuando sales llorando de un examen”. De felicidad, de odio, de rabia o de tristeza, tú siempre me esperaste, siempre me abrazaste y me distes toda la fuerza que me faltaba, toda la confianza que no tenía en mí. Juntos hemos descubierto mucho más que la escuela porque siempre me has acompañado. Te quiero en Roma frente al Coliseo, en París sentados frente a “La Libertad Guiando al Pueblo” bajo la pirámide del Louvre, en Nueva York viendo anochecer desde Brooklyn, en Londres de compras, en Brujas paseando por sus canales, en Miami bajo el sol y brindando por nosotros, en Islandia bailando bajo tormentas, volcanes, glaciares y cascadas. Te quiero en el resto de lugares que hemos visitado juntos, por los que hemos paseado y marcado nuestra vida llenando de recuerdos nuestra alma. Pero sobre todo te quiero por nuestro presente y nuestro futuro. Sea donde sea. Sea como sea. Juntos.

ESCUELA SUPERIOR DE INGENIEROS, SEVILLA

Susana Hornillo. Cuando se realiza con gusto, el resultado no puede ser más óptimo. Gran profesora y persona. La dedicación puesta en este proyecto, los ánimos y cada palabra para mejorar y para crecer las guardo en una caja muy dentro de mí, para siempre.

La fortuna no es una ruleta.

Estar rodeada cada día de personas que te hacen sentir afortunada es realmente por lo que quiero dar gracias. Gracias a cada uno de los nombrados anteriormente por ser como sois, y por serlo a mi lado.

Gracias desde el día uno,
por caminar junto a mí,
por dar la mano a un amigo,
por permitirme ser lo que quiero ser,
y aun así quererme.
Gracias por demostrarme tanto.
Jamás os podré devolver lo recibido.
Gracias hasta el final.
Gracias por ser parte del inicio del fin.

*María Laó Cañadas
Sevilla, 2018*

Resumen

Basándose en la recomendación de la ITU-R P.526 este proyecto explica distintos métodos matemáticos para tratar radioenlaces que presentan difracción en su recorrido. Se hace distinción según el tipo de medio que se está tratando, medio llano o con obstáculos y dentro de este último se adentrará en la forma de resolución para uno y dos obstáculos aislados, tanto filo de cuchillo como con radio conocido.

Una vez resuelto los conceptos clave de los que dependen el cálculo de las pérdidas, se realiza un código en Python que implementa lo explicado con anterioridad, es decir, se presenta un programa capaz de calcular las pérdidas en medios llanos, con uno y dos obstáculos simplemente introduciendo algunos parámetros que configura el usuario como la frecuencia, el radio efectivo de la tierra o la altura de los mástiles de las antenas, además de las coordenadas (latitud y longitud en formato decimal) dónde se quieren colocar las antenas, transmisora y receptora. Con estos últimos datos, y haciendo uso de una API de Google, cuyo uso se ha adaptado a Python también en este trabajo, se consiguen otros tales como la distancia entre antenas, la distancia a posibles obstáculos, la elevación del terreno en cada uno de los puntos en los cuales se analiza, despejamiento y otros factores de carácter imprescindible en nuestro diseño del radioenlace.

Para verificar el correcto funcionamiento del programa propuesto, se han realizado una serie de problemas ejemplo con resolución analítica y su posterior comparación con los datos que nos muestra por pantalla el código al ser ejecutado introduciendo coordenadas reales obtenidas con Google Earth, para así poder afirmar que todo resulta como lo indicado en la norma a la cual se hace referencia en este proyecto fin de grado.

Abstract

Based on ITU-R P.526 this project explains several mathematical methods to solve radio links that present diffraction in their path. It is about that the type of terrain that we are dealing with, smooth terrain or isolated obstacles (one or two) with the knife-edge obstacle or with rounded obstacles method.

Once the key concepts have been understood, the project continues with a Python code that implements the previously explained. This is a program capable of calculating the radio link losses simply entering some parameters that configure the user as the frequency, the effective radius of the earth or the height of the antennas, in addition, you can decide the coordinates (latitude and longitude in decimal format) where you want to place the antennas, transmitter and receiver. With these data, and making use of a Google API, whose use is adapted to Python also in this work, other parameters have been calculated, such as the distance between the antennas, the distance to possible obstacles, the elevation of the analyzed terrain in each one of the points and other indispensable factors in our design of the radio link.

To verify that the program works correctly, a series of problems has been made with the analytical test and its subsequent comparison with the data shown by the screen when the code is executed with real coordinates taken with Google Earth. All of this is made to be able to affirm that everything turns out as indicated in the norm which is reference in this final degree project.

Índice

Agradecimientos	I
Resumen	III
Abstract	V
Índice	VII
Índice de Figuras	IX
Notación	XI
1 Introducción	1
1.1 <i>Motivación</i>	1
1.2 <i>Objetivos</i>	2
2 Conceptos Básicos	3
2.1 <i>Elipsoides de Fresnel</i>	3
2.2 <i>Zona de penumbra</i>	4
2.3 <i>Zona de difracción</i>	5
2.4 <i>Tipos de terreno</i>	5
2.4.1 Terreno llano	5
2.4.2 Terreno con obstáculos aislados	5
2.4.3 Terreno ondulado	5
2.5 <i>Integrales de Fresnel</i>	6
3 Propagación por Difracción (ITU-R P.526-13)	7
3.1 <i>Medio llano</i>	7
3.1.1 Difracción sobre el horizonte	7
3.1.2 Procedimiento de interpolación lineal	10
3.2 <i>Medio con obstáculos</i>	11
3.2.1 Un único obstáculo aislado	12
3.2.2 Dos obstáculos aislados	15
4 Resolución Analítica	19
4.1 <i>Medio Llano</i>	19
4.1.1 Caso 1: Distancia del enlace menor a d_{los} , $h < h_{req}$ y $1 > K > 0.001$	19
4.1.2 Caso 2: Distancia del enlace menor a d_{los} y $h > h_{req}$	23
4.2 <i>Medio con Obstáculos</i>	25
4.2.1 Caso 1: Un obstáculo “Filo de Cuchillo”	25
4.2.2 Caso 2: Un obstáculo con radio conocido	27
4.2.3 Caso 3: Dos obstáculos aislados “Filo de Cuchillo”	28
4.2.4 Caso 4: Dos obstáculos aislados con radios conocidos	31
4.2.5 Caso 5: Obstáculo que hace inviable el radioenlace	33
5 Programa en Python	35
5.1 <i>Programa principal: TFG_Principal.py</i>	35
5.2 <i>Funciones utilizadas: Terreno.py</i>	37
5.2.1 <code>perfil_terreno(laTX, lonTX, latRX, lonRX, muestras)</code>	37

5.2.2	terrain_type(dis, elevacion, wave_len, a_e, h_TX, h_RX, latitud, longitud, muestras)	37
5.3	<i>Funciones utilizadas: apiTFG.py</i>	37
5.3.1	getElevation(path, samples, sensor="false", **elvt_n_args)	37
5.3.2	GetDistance(lat1, lon1, lat2, lon2, R)	37
5.4	<i>Funciones utilizadas: TerrenoLlano.py</i>	38
5.4.1	medio_llano(a_e, h_TX, h_RX, dis, wave_len, freq)	38
5.4.2	admitancia_superficie(wave_len, a_e)	38
5.4.3	fuerza_difraccion_campo(K, dis, wave_len, a_e, freq)	38
5.4.4	calc_beta(K, tierra_o_mar, freq)	39
5.4.5	calc_G(Y, beta, K)	39
5.4.6	h_curvatura_rayo(dis, h_TX, h_RX, a_e, wave_len)	39
5.4.7	perdidas(h, d_TX, d_RX, wave_len, dis)	39
5.5	<i>Funciones utilizadas: TerrenoObstaculos.py</i>	39
5.5.1	numero_obstaculos(dis, latitud, longitud, elevacion, wave_len, a_e, muestras)	39
5.5.2	unico_obs(wave_len, dis, d1, d2, despejamiento)	40
5.5.3	calc_Jv(v)	40
5.5.4	calc_Tmn(r, wave_len, d1, d2, h)	40
5.5.5	dos_obs(wave_len, dis, d1, d2, despejamiento, elev, lat, lon, a_e, h_RX)	40
5.5.6	calc_Tc(wave_len, d1, d2, dis_obs, despejamiento, h_nueva)	41
6	Comparación de Resultados	43
6.1	<i>Medio Llano</i>	43
6.1.1	Caso 1: Distancia del enlace menor a d_{los} y $1 > K > 0.001$	43
6.1.2	Caso 2: Distancia del enlace menor a d_{los} y $h > h_{req}$	45
6.2	<i>Medio con obstáculos</i>	46
6.2.1	Caso 1: Un obstáculo "Filo de Cuchillo"	46
6.2.2	Caso 2: Un obstáculo con radio conocido	48
6.2.3	Caso 3: Dos obstáculos aislados "Filo de Cuchillo"	49
6.2.4	Caso 4: Dos obstáculos aislados con radios conocidos	50
6.2.5	Caso 5: Obstáculo que hace inviable el radioenlace	50
7	Conclusiones	53
	Referencias	55
	Glosario	57
	Anexo I	59
	Anexo II	61
	Anexo III	65
	<i>Solicitudes de elevación</i>	65
	<i>Parámetros de la solicitud</i>	65
	<i>Respuestas de elevación</i>	66
	<i>Obtener una clave de autenticación</i>	66
	<i>Límites que se aplican a Google Maps Elevation API con uso estándar</i>	67
	Anexo IV	69
	<i>TFG_Principal.py</i>	69
	<i>Terreno.py</i>	73
	<i>TerrenoLlano.py</i>	75
	<i>TerrenoObstaculos.py</i>	79
	<i>apiTFG.py</i>	83
	<i>Cv_Sv.py</i>	85

ÍNDICE DE FIGURAS

Figura 2-1. Ejemplo zonas de Fresnel. [3]	4
Figura 2-2. Ancho de la zona de penumbra. [3].....	5
Figura 3-1. Definición gráfica del parámetro h . [7]	10
Figura 3-2. Obstáculo filo de cuchillo – Caso 1. [7].....	12
Figura 3-3. Obstáculo filo de cuchillo – Caso 2. [8].....	13
Figura 3-4. Representación de un obstáculo aislado con radio conocido y de altura h . [8].....	14
Figura 3-5. Pérdidas obstáculo filo de cuchillo. [9].....	15
Figura 3-6. Esquema de un radioenlace con dos obstáculos aislados – Filo de cuchillo. [10]	15
Figura 3-7. Esquema de un radioenlace con dos obstáculos aislados dónde uno predomina – Filo de cuchillo. [11].....	16
Figura 4-1. Esquema del radioenlace propuesto – Caso 1 Sin Obstáculos.	20
Figura 4-2. Esquema del radioenlace propuesto – Caso 2 Sin Obstáculos.	23
Figura 4-3. Esquema del radioenlace propuesto – Caso 1 Con Obstáculos.	25
Figura 4-4. Esquema del radioenlace propuesto – Caso 3 Con Obstáculos.	28
Figura 4-5. Esquema del radioenlace propuesto – Caso 5 Con Obstáculos.	33
Figura 5-1. Diagrama de flujo del programa.	36
Figura 6-1. Enlace y perfil elevación Google Earth.....	44
Figura 6-2. Ejecución Enlace 6-1.	44
Figura 6-3. Enlace y perfil elevación Google Earth.....	45
Figura 6-4. Ejecución Enlace 6-3.	46
Figura 6-5. Enlace y perfil elevación Google Earth.....	47
Figura 6-6. Ejecución Enlace 6-5.	47
Figura 6-7. Enlace y perfil elevación Google Earth.....	48
Figura 6-8. Ejecución Enlace 6-7.	48
Figura 6-9. Enlace y perfil elevación Google Earth.....	49
Figura 6-10. Ejecución Enlace 6-9.	50
Figura 6-11. Ejecución Enlace 6-9.	50
Figura 6-12. Enlace y perfil elevación Google Earth.....	51
Figura 6-13. Ejecución Enlace 6-12.	51

Notación

TX	Transmisor
RX	Receptor
IRe	Parte real
IIm	Parte imaginaria
sen	Función seno
tanh	Función tangente hiperbólica
arctg	Función arco tangente
sen	Función seno
<	Menor
>	Mayor
\leq	Menor o igual
\geq	Mayor o igual
°	Grados

1 INTRODUCCIÓN

*A TI, donaire alado, forma en vuelo,
Raudo volumen que la luz reanima
y en el movible espacio determina
la paralela sombra de su anhelo.*

- Rafael Alberti -

Las bases teóricas de la propagación de ondas electromagnéticas de James C. Maxwell fueron el principio del cambio. Gracias a sus demostraciones matemáticas y ecuaciones propuestas fue el humano adentrándose en el mundo de las telecomunicaciones basadas en radiocomunicación.

Hoy en día, esta es la base de todo, radio, televisión o telefonía móvil no serían nada sin el uso de antenas, transmisores y receptores capaces de enviar, recibir e interpretar las ondas que viajan a nuestro alrededor.

Por otro lado, Python es un lenguaje de programación cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje muy potente y multiparadigma ya que soporta orientación a objetos, programación imperativa y programación funcional. Además, posee licencia de código abierto compatible con diversas versiones de GNU.

Si unimos ambos conceptos se pueden conseguir grandes objetivos, como el que se proponen en este proyecto.

1.1 Motivación

Los conceptos de radioenlaces aprendidos a lo largo de estos años en el grado han sido en gran medida los que han hecho que la radiocomunicación sea la parte que más he disfrutado aprendiendo.

Poder transmitir este gusto por algo en forma de un trabajo que represente el fin en esta escuela, es como dejar parte de lo que me ha gustado para que, en el futuro, alguien pueda sentir ese mismo orgullo al realizar un trabajo basado en radiocomunicación con ayuda de este.

No solo buscaba proyectar los conocimientos adquiridos, quería ir algo más allá. La programación es indispensable para el presente y para el futuro, y apostar por un lenguaje robusto y en crecimiento constante era justo lo que necesitaba para decidirme por hacerlo. Aprender Python [1] se ha convertido en uno de los retos que se proponen en este trabajo. Finalmente, y como punto final a esta motivación, solo añadir que el simple hecho de pensar que en algún momento futuro alguien pueda hacer uso de este trabajo junto con otros para dar

prácticas de asignaturas que yo misma cursé es lo que realmente ha conseguido que estas palabras que vienen a continuación puedan ser escritas.

1.2 Objetivos

Los objetivos principales de este trabajo fin de grado son los que se indican a continuación:

- Comprender los métodos de resolución de enlaces en medio llano, un obstáculos y dos obstáculos aislados basados en la recomendación de la ITU-R P.526.
- Obtener conocimientos de programación en Python para poder codificar funciones de diferentes dificultades que presenten propiedades como pueden ser el pedir parámetros por pantalla, que estos se guarden, imprimir resultados y o el codificar operaciones matemáticas.
- Realizar un código que implemente la parte de la recomendación explicada utilizando parámetros que configura el usuario.
- Ampliar el código para que pueda ser lo más realista posible, en este caso con la obtención del perfil del terreno de forma real solo indicando las posiciones en las que se encuentra la antena transmisora y la receptora.
- Comparar resultados analíticos con reales.

2 CONCEPTOS BÁSICOS

*¿De dónde vengo? El más horrible y áspero
de los senderos busca;
las huellas de unos pies ensangrentados
sobre la roca dura;*

*¿Adónde voy? El más sombrío y triste
de los páramos cruza,
valle de eternas nieves y de eternas
melancólicas brumas;*

- Gustavo Adolfo Bécquer -

La recomendación en la cual se basa este proyecto [2] presenta unos modelos dedicados al estudio del efecto de la difracción en la intensidad del campo recibido en un radioenlace.

La difracción que se estudia es aquella producida por la propia superficie y curvatura de la tierra o por otros obstáculos. En la resolución de este tipo de problemas se tiene que tener en cuenta fenómenos como la refracción atmosférica en la transmisión para el cálculo de determinados parámetros como pueden ser el ángulo de difracción o el radio de curvatura.¹

2.1 Elipsoides de Fresnel

Las zonas de Fresnel son unos elipsoides concéntricos que rodean al rayo directo de un enlace radioeléctrico entre dos puntos, antena transmisora (punto A) y antena receptora (punto B).

Si tomamos un punto, M, perteneciente a cualquiera de los elipsoides, como el lugar donde se refleja el rayo que proviene de la antena transmisora y tras esto, incide sobre la antena receptora este habrá recorrido una distancia superior a la recorrida por el rayo directo en múltiplos de media longitud de onda. Es decir, la onda reflejada se recibirá con un retardo respecto al rayo directo, equivalente a un desfase múltiplo de 180°. El valor del múltiplo es lo que dará lugar al n-ésimo elipsoide de Fresnel.

$$AM + BM = AB + n \frac{\lambda}{2} \quad (2-1)$$

Posibles reflexiones cerca del borde de la primera zona de Fresnel ($n = 1$) pueden causar atenuación, ya que la onda reflejada llegaría a la antena receptora en oposición de fase. Por lo tanto, debe asegurarse que la primera zona de Fresnel se encuentre libre de obstáculos (LoS) pues una obstrucción puede provocar grandes pérdidas en la transmisión. Si no lo está, tendría que buscarse una solución como aumentar la altura de los mástiles de las antenas o situarlas en otra posición.

¹ Si no se dan otros datos, se tomará el radio equivalente de la tierra por 8500km

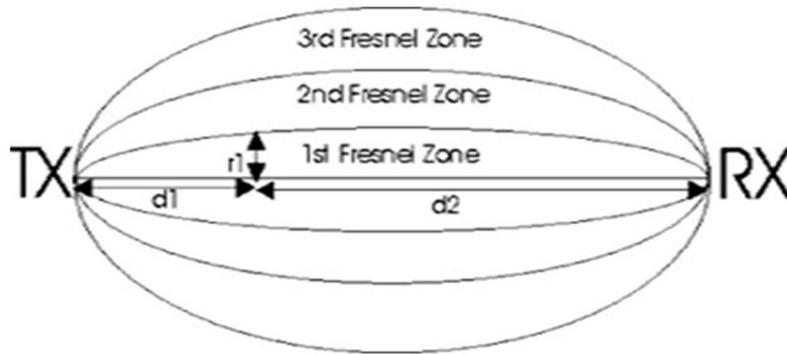


Figura 2-1. Ejemplo zonas de Fresnel. [3]

El radio de cada una de las zonas de Fresnel en un punto determinado se puede calcular utilizando la siguiente expresión:

$$R_n = \sqrt{\frac{n\lambda d_1 d_2}{d_1 + d_2}} \quad m \quad (2-2)$$

Siendo:

λ : longitud de onda (m)

d_1 : distancia desde el transmisor hasta el punto donde se está midiendo el radio (m)

d_2 : distancia desde donde se está midiendo el radio hasta el receptor (m)

Para obtener el radio máximo de la n-ésima zona de Fresnel se tiene que medir en el punto medio entre la antena transmisora y la antena receptora.

Hay que tener en cuenta que a medida que la frecuencia aumenta los elipsoides se hacen cada vez más estrechos.

2.2 Zona de penumbra

La región o zona de penumbra es aquella que se define como la transición entre la luz y la sombra. Esta transición es una fina línea (ancho de la penumbra) en el límite del elipsoide donde empieza la sombra.

La ecuación que nos da el ancho (w) en el caso de un transmisor que se encuentra a una altura h , sobre una tierra suave y esférica es:

$$w = \sqrt[3]{\frac{\lambda a_e^2}{\pi}} \quad m \quad (2-3)$$

Dónde:

λ : longitud de onda (m)

a_e : radio efectivo de la tierra (m)

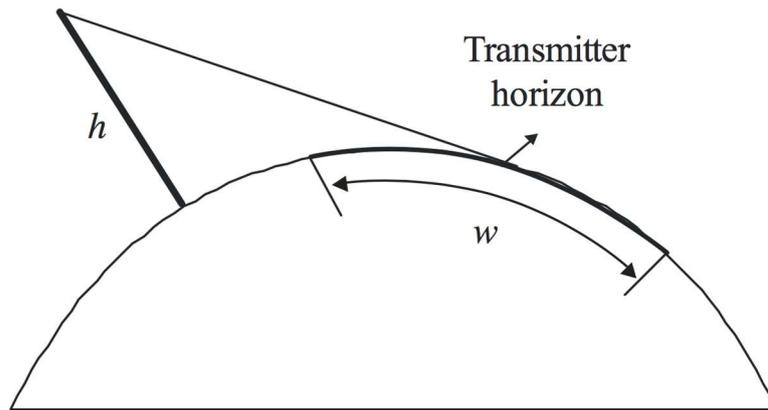


Figura 2-2. Ancho de la zona de penumbra. [3]

2.3 Zona de difracción

La teoría de la difracción indica que la ruta directa entre el transmisor y el receptor necesita un espacio libre de al menos el 60% del radio de la primera zona de Fresnel para lograr condiciones de propagación en el espacio libre, LoS.

La zona de difracción de un transmisor, por tanto, se extiende desde la distancia LoS hasta una distancia mucho más allá del horizonte del transmisor donde el mecanismo de la dispersión troposférica se vuelve predominante.

2.4 Tipos de terreno

En la superficie terrestre existen obstáculos que tienen irregularidades de una determinada altura que se define como Δh y se calcula como sigue:

$$\Delta h = 0.04 \sqrt[3]{R\lambda^2} \quad m \quad (2-4)$$

Dependiendo del valor de este parámetro se pueden clasificar los tipos de terreno en tres, terreno llano, terreno con obstáculos aislados y terreno ondulado.

2.4.1 Terreno llano

Si se considera R el valor máximo del radio en la primera zona de Fresnel, se podrá considerar que estamos trabajando sobre una tierra plana cuando las irregularidades del terreno son inferiores o del orden de $0.1R$.

2.4.2 Terreno con obstáculos aislados

Cuando se estudia este tipo de terreno se pueden encontrar uno o más obstáculos aislados que afectan a la propagación.

2.4.3 Terreno ondulado

El perfil que se presenta es aquel formado por numerosas colinas pequeñas de las cuales ninguna es dominante frente a las demás.

2.5 Integrales de Fresnel

La integral compleja de Fresnel viene dada por:

$$F_c(v) = \int_0^v e^{j\frac{\pi s^2}{2}} ds = C(v) + jS(v) \quad (2-5)$$

Dónde:

$$C(v) = \int_0^v \cos\left(\frac{\pi s^2}{2}\right) ds \quad (2-6)$$

$$S(v) = \int_0^v \sin\left(\frac{\pi s^2}{2}\right) ds \quad (2-7)$$

Más adelante, sólo se tomará el primer término de la serie para hacer aproximaciones.

3 PROPAGACIÓN POR DIFRACCIÓN (ITU-R P.526-13)

*Las más hondas palabras
del sabio nos enseñan
lo que el silbar del viento cuando sopla
o el sonar de las aguas cuando ruedan.*

- Antonio Machado -

Gracias a los conceptos anteriores se puede comenzar a explicar la resolución de un radioenlace con presencia de difracción.

Lo primero que se tiene que tener claro es el terreno en el cual se encuentran tanto el transmisor y el receptor, como el radioenlace que los une.

3.1 Medio llano

Una vez concluido con que nos encontramos en un medio llano, procedemos a analizar la línea de visión directa. Como se indicó en el punto anterior, si el 60% del primer elipsoide de Fresnel se encuentra libre de obstáculos se puede decir que si tenemos una conexión entre TX y RX sin presencia de obstáculos que produzcan pérdidas. En este caso, y para frecuencias mayor o igual a 10MHz y para un radio efectivo de la tierra mayor que cero, $a_e > 0$, se calcula el margen de distancia LoS que viene dado por:

$$d_{los} = \sqrt[2]{2a_e}(\sqrt[2]{h_1} + \sqrt[2]{h_2}) \quad m \quad (3-1)$$

Si $d \geq d_{los}$ se utiliza para calcular las perdidas el método de difracción sobre el horizonte.

Si $d \leq d_{los}$ se tendrá que realizar un procedimiento de interpolación basado en un a_e distinto del introducido.

3.1.1 Difracción sobre el horizonte

Este tipo de difracción para largas distancias solo tiene en cuenta el primer término de la serie, en cambio si la longitud no es demasiado grande o es el propio horizonte, esta aproximación solo es válida si el error es como máximo 2dB.

El cálculo de ese primer término se puede realizar de forma numérica o mediante el uso de nomogramas.

3.1.1.1 Cálculo numérico

Las características eléctricas de la Tierra pueden influenciar a las pérdidas por difracción, las cuales, son determinadas con el cálculo de un factor normalizado denominado admitancia de superficie, K [5], y que tiene en cuenta la polarización.

Para polarización horizontal:

$$K_H = \left(\frac{2\pi a_e}{\lambda} \right)^{-1/3} [(\varepsilon - 1)^2 + (60\lambda\sigma)^2]^{-1/4} \quad (3-2)$$

Y para polarización vertical:

$$K_V = K_H \sqrt{\varepsilon^2 + (60\lambda\sigma)^2} \quad (3-3)$$

Donde:

a_e : radio efectivo de la tierra (km)

ε : permitividad relativa (elegida de la gráfica del ANEXO I)

σ : conductividad (S/m) (elegida de la gráfica del ANEXO I)

Si K es menor que 0.001 las características eléctricas de la tierra serán despreciables y si es mayor que 1 habría que utilizar un procedimiento especial que ya está implementado en un programa de la ITU-R llamado GRWAVE. Este valor mayor que uno solo ocurre para polarización vertical cuando la frecuencia a la que se está trabajando es menor de 10MHz sobre mar o menor a 200KHz sobre tierra.

Para el caso $0.001 < K < 1$ hay que continuar calculando la fuerza de difracción del campo, E , que es relativa a la que tiene en el espacio libre E_0 y viene dada por:

$$20 \log \frac{E}{E_0} = F(X) + G(Y_1) + G(Y_2) \quad dB \quad (3-4)$$

El término logarítmico suele ser negativo, es decir E es menor que E_0 , por lo general.

F es un término que depende de la distancia, X la longitud normalizada del terreno entre ambas antenas que tienen alturas, también normalizadas, Y_1 e Y_2 y G [6] la ganancia de altura.

$$F(X) = \begin{cases} -20 \log(X) - 5.6488X^{1.425} & X < 1.6 \\ 11 + 10 \log(X) - 17.6X & X \geq 1.6 \end{cases} \quad (3-5)$$

$$X = \beta d \sqrt[3]{\frac{\pi}{\lambda a_e^2}} \quad (3-6)$$

$$G(Y) \cong \begin{cases} 20 \log(B + 0.1B^3) & B \leq 2 \\ 17.6 \sqrt[2]{B - 1.1} - 5 \log(B - 1.1) & B > 2 \end{cases} \quad (3-7)$$

En el caso de que $G(Y) < 2 + 20 \log K$ el valor que tomará será ese. Para el resto se necesita B que se tomará como $B = \beta Y$.

$$Y = 2\beta h \sqrt[3]{\frac{\pi^2}{\lambda^2 a_e}} \quad (3-8)$$

d : longitud del terreno (m)

h : altura de la antena (m)

β : parámetro que depende del tipo de terreno y de la polarización. Está relacionado con K .

Cuando la polarización es horizontal (para todas las frecuencias) o la polarización es vertical (con frecuencias mayores a 20MHz sobre tierra o 300MHz sobre mar), β se toma con valor igual a 1. Para el resto de las frecuencias tiene que calcularse en función de K . Una buena aproximación puede realizarse utilizando la expresión que se indica:

$$\beta = \frac{1 + 1.6K^2 + 0.67K^4}{1 + 4.5K^2 + 1.53K^4} \quad (3-9)$$

Todas estas aproximaciones se han realizado teniendo en cuenta solo el primer término de la serie residual, pero solo son válidas (es decir, no difieren más de $2dB$) si se cumple:

$$X - \sqrt[2]{\beta Y_1} \Delta(Y_1, K) - \sqrt[2]{\beta Y_2} \Delta(Y_2, K) > X_{lim} \quad (3-10)$$

Donde:

$$X_{lim} = 1.096 - 1.280(1 - \beta) \quad (3-11)$$

$$\Delta(Y, K) = \Delta(Y, 0) + 1.779(1 - \beta)[\Delta(Y, \infty) - \Delta(Y, 0)] \quad (3-12)$$

$$\Delta(Y, 0) = 0.5 \left[1 + \tanh \left(\frac{0.5 \log \beta Y - 0.255}{0.3} \right) \right] \quad (3-13)$$

$$\Delta(Y, \infty) = 0.5 \left[1 + \tanh \left(\frac{0.5 \log \beta Y + 0.255}{0.25} \right) \right] \quad (3-14)$$

Además, la distancia mínima para que se pueda usar la ecuación (3-4) tiene que cumplir:

$$X_{min} = X_{lim} + (\beta Y_1)^{1/2} \Delta(Y_1, K) + (\beta Y_2)^{1/2} \Delta(Y_2, K) \quad (3-15)$$

Usando esta X_{min} en la ecuación (2-6).

3.1.1.2 Cálculo con nomogramas

Basados en la misma aproximación que la resolución por cálculo numérico, se puede realizar con el uso de nomogramas que darán los valores para completar esta ecuación:

$$20 \log \frac{E}{E_0} = F(d) + H(h_1) + H(h_2) \quad (3-16)$$

Los nomogramas dan el valor recibido de las funciones F y H en espacio libre para valores de $k=1$ y $k=4/3$. Además la frecuencia tiene que ser mayor a 30MHz.

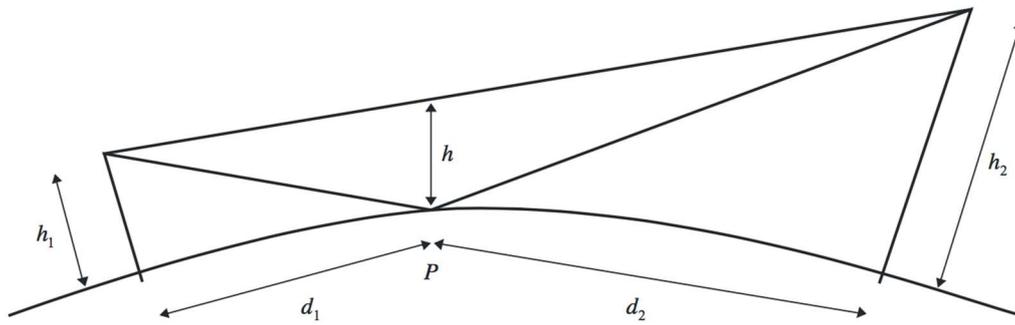
Estos nomogramas también se pueden usar para otros valores de k haciendo un cambio en la frecuencia a la cual se está trabajando. Concretamente, habría que realizar:

$$f = \frac{f}{k^2} \quad \text{para las Figs 1 y 2 del anexo 1} \quad (3-17)$$

$$f = \frac{f}{\sqrt[2]{k}} \quad \text{para las Figs 2 y 3 del anexo 1} \quad (3-18)$$

3.1.2 Procedimiento de interpolación lineal

Una vez determinado d_{los} hay que pasar a calcular cuál es el valor más pequeño de altura desde la curvatura de la tierra hasta el rayo que viaja entre ambas antenas, este parámetro se denomina h .



P: Reflection point

P.0526-07

Figura 3-1. Definición gráfica del parámetro h . [7]

$$d_1 = \frac{d}{2}(1 + b) \quad (3-19)$$

$$d_2 = d - d_1 \quad (3-20)$$

$$h = \frac{\left(h_1 - \frac{d_1^2}{2a_e}\right)d_2 + \left(h_2 - \frac{d_2^2}{2a_e}\right)d_1}{d} \quad (3-21)$$

Donde:

$$b = 2 \sqrt{\frac{m+1}{3m}} \cos \left(\frac{\pi}{3} + \frac{1}{3} \arccos \left(\frac{3c}{2} \sqrt{\frac{3m}{(m+1)^3}} \right) \right) \quad (3-22)$$

$$m = \frac{d^2}{4a_e(h_1 + h_2)} \quad (3-23)$$

$$c = \frac{h_1 - h_2}{h_1 + h_2} \quad (3-24)$$

Para que las pérdidas por difracción no afecten al radioenlace se tiene que cumplir que el valor de h calculado sea mayor que el mínimo h_{req} para el cual no existen pérdidas.

Siendo:

$$h_{req} = 0.552 \sqrt{\frac{d_1 d_2 \lambda}{d}} \quad (3-25)$$

Para el caso en que no se cumple esa condición, hay que calcular un radio efectivo de la tierra modificado (a_{em}) que nos dará un margen de LoS para d mediante:

$$a_{em} = 0.5 \left(\frac{d}{\sqrt[2]{h_1} + \sqrt[2]{h_2}} \right)^2 \quad (3-26)$$

Tras esto, se utiliza el método explicado en el apartado anterior, “*Difracción sobre el horizonte*”, para calcular las pérdidas sustituyendo a_e por a_{em} y asignando esas pérdidas a A_h .

Si A_h es negativo, las pérdidas por difracción serán cero.

Si no lo es, se calcula la difracción interpolada, $A(dB)$ y concluiremos el estudio.

$$A = \left[1 - \frac{h}{h_{req}} \right] A_h \quad (3-27)$$

3.2 Medio con obstáculos

En numerosas ocasiones, se encuentran terrenos que presentan uno o más obstáculos por lo que las pérdidas que estos producen tienen que ser estimadas para conseguir un cómputo total lo más fiable posible.

Para que un obstáculo sea considerado “aislado” tiene que cumplir las siguientes características:

- No puede haber interacción entre el obstáculo y el medio que lo rodea.
- No debe existir superposición entre la zona de penumbra asociada a cada antena y la parte superior del obstáculo.
- El terreno libre a ambos lados del obstáculo tiene que ser, al menos, un 60% del radio de la primera zona de Fresnel.
- La reflexión especular no puede presentarse a ambos lados del obstáculo.

Todos los cálculos que se van a proponer según los obstáculos son para formas idealizadas de estos ya que nos permite hacer una buena aproximación de las pérdidas reales. La documentación que se va a dar a partir de ahora puede aplicarse cuando la longitud de onda es lo suficientemente pequeña en comparación con el tamaño del obstáculo.

Cuando nos encontramos en un medio de este tipo, tenemos que tener varias cosas en cuenta, la primera de ellas es cuántos obstáculos aislados están presentes en el radioenlace y la segunda si el radio del obstáculo es conocido o no. Tras esto, se continuará con un modo de resolución particular.

3.2.1 Un único obstáculo aislado

Tal y como se ha indicado anteriormente, si el radio del obstáculo es conocido o no, se procederá de forma distinta.

Cuando el radio no es conocido tendremos lo que se denomina obstáculo “filo de cuchillo” que es un caso extremadamente idealizado y no reflexivo, en el que todos los parámetros están combinados en uno único, v , que difiere dependiendo los parámetros geométricos elegidos. v puede calcularse, en este caso, utilizando alguna de las fórmulas que siguen:

$$v = \theta^2 \sqrt{\frac{2}{\lambda \left(\frac{1}{d_1} + \frac{1}{d_2} \right)}} \quad (3-28)$$

$$v = \frac{2h\theta}{\lambda} \quad (3-29)$$

$$v = \frac{2d}{\lambda} \alpha_1 \alpha_2 \quad (3-30)$$

$$v = h^2 \sqrt{\frac{2}{\lambda} \left(\frac{1}{d_1} + \frac{1}{d_2} \right)} \quad (3-31)$$

Dónde:

h : altura que va desde la línea recta que une el TX y el RX hasta la parte más alta del obstáculo (m). Si la altura está por debajo de esta línea, h será negativo.

d_1 : distancia desde el TX al top del obstáculo (m).

d_2 : distancia desde el top del obstáculo al RX (m).

d : distancia desde el TX al RX (m).

θ : ángulo de difracción (rad). Tiene el mismo signo que h .

α_1 : ángulo que se forma entre la línea que une la parte alta del obstáculo con el TX con la línea recta que une el TX con el RX.

α_2 : ángulo que se forma entre la línea que une la parte alta del obstáculo con el RX con la línea recta que une el TX con el RX.

En la ecuación (2 - 28) v tiene el signo de h y de θ y en la ecuación (2 - 30) el de α_1 y α_2 .

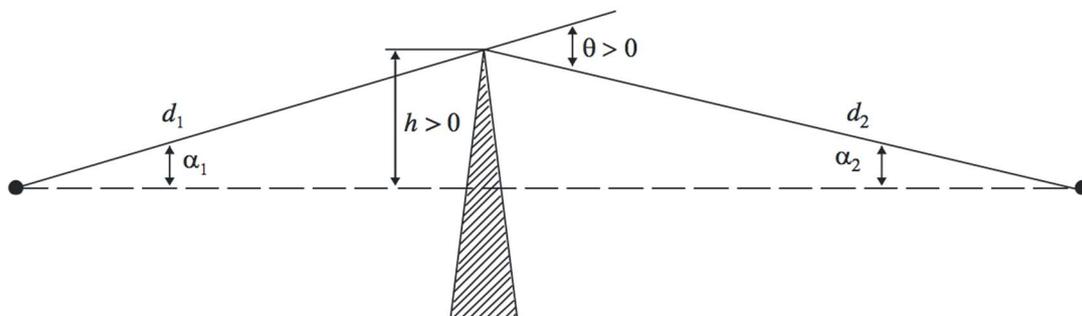


Figura 3-2. Obstáculo filo de cuchillo – Caso 1. [7]

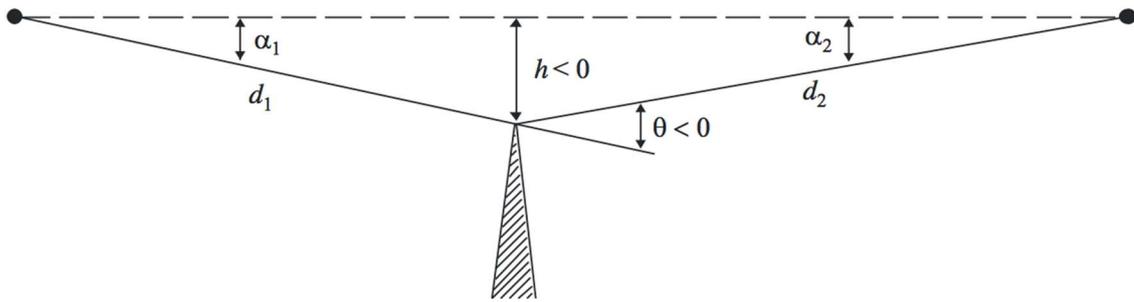


Figura 3-3. Obstáculo filo de cuchillo – Caso 2. [8]

Se puede comprobar que tenemos dos modelos iguales, pero con el valor de un parámetro distinto, el de h . Esto da lugar a que en determinados enlaces encontremos un despejamiento negativo ($h < 0$).

Para el cálculo de las pérdidas se utiliza la función $J(v)$ que depende de $C(v)$ y $S(v)$ que son las partes real e imaginaria de las integrales complejas de Fresnel.

$$J(v) = -20 \log \left(\frac{\sqrt{[1 - C(v) - S(v)]^2 + [C(v) - S(v)]^2}}{2} \right) \quad (3-32)$$

Si v es mayor que -0.78 esta fórmula puede simplificarse, dando la expresión:

$$J(v) = 6.9 + 20 \log \left(\sqrt{(v - 0.1)^2 + 1} + v - 0.1 \right) \quad (3-33)$$

Además, se puede utilizar la gráfica de la figura 2-5 para relacionar directamente $J(v)$ con v . Con esta se puede observar que la atenuación desaparece cuando el despejamiento es igual al 60% del radio de la primera zona de Fresnel, criterio que suele utilizarse en la práctica para el diseño de radioenlaces.

En el caso de que sí tengamos el parámetro R y contemos con otros datos como d_1 , d_2 (medidas desde el vértice donde el rayo se proyecta encima del obstáculo) y la altura sobre la línea base, h , tendremos un perfil como el que se presenta en la figura 2-4. Las pérdidas por difracción pueden ser calculadas como:

$$A = J(v) + T(m, n) \quad (3-34)$$

Se tiene en cuenta que $J(v)$ son las pérdidas Fresnel-Kirchoff producidas por un equivalente “*filo de cuchillo*”, modelo explicado anteriormente y, del cual se pueden obtener las fórmulas de v y de $J(v)$, pues, en ambos casos coinciden. La fórmula que suele usarse es la ecuación (2-33), previamente dada ya que v es positivo y por tanto la línea de visión directa está obstruida por el obstáculo.

$J(v)$ puede obtenerse también utilizando la figura 2-5.

Por otro lado, $T(m, n)$ es la atenuación adicional dada por la curvatura del obstáculo y viene definida como:

$$T(m, n) = \begin{cases} 7.2\sqrt[3]{m} - (2 - 12.5n)m + 3.6m^{3/2} - 0.8m^2 & mn \leq 4 \\ -6 - 20 \log(mn) + 7.2\sqrt[3]{m} - (2 - 17n)m + 3.6m^{3/2} - 0.8m^2 & mn > 4 \end{cases} \quad (3-35)$$

Dónde:

$$m = \frac{R \frac{(d_1 + d_2)}{d_1 d_2}}{\sqrt[3]{\frac{\pi R}{\lambda}}} \quad (3-36)$$

$$n = \frac{h \left(\frac{\pi R}{\lambda}\right)^{2/3}}{R} \quad (3-37)$$

h : altura que va desde la línea recta que une el TX y el RX hasta la parte más alta del obstáculo (m).

d_1 : distancia desde el TX al top del obstáculo (m).

d_2 : distancia desde el top del obstáculo al RX (m).

R : radio del obstáculo, es conocido (m).

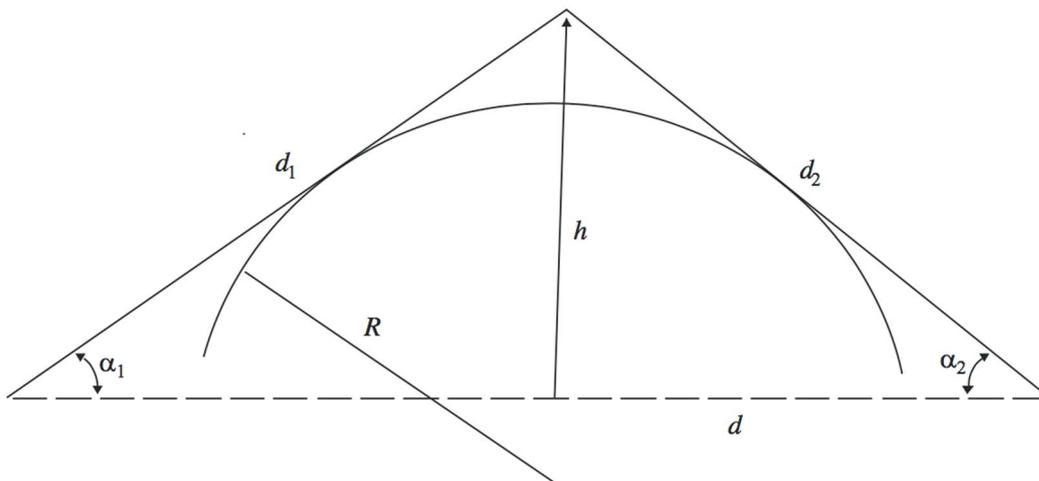


Figura 3-4. Representación de un obstáculo aislado con radio conocido y de altura h . [8]

El radio de curvatura del obstáculo se corresponde con el existente en el ápice de una parábola, que está ajustada al perfil de obstáculo en las proximidades de la parte superior. Para ajustar dicha parábola, hay que tener en cuenta que la máxima distancia vertical, desde la parte alta, utilizada, debe de ser del orden del radio del primer elipsoide de Fresnel del radioenlace en estudio.

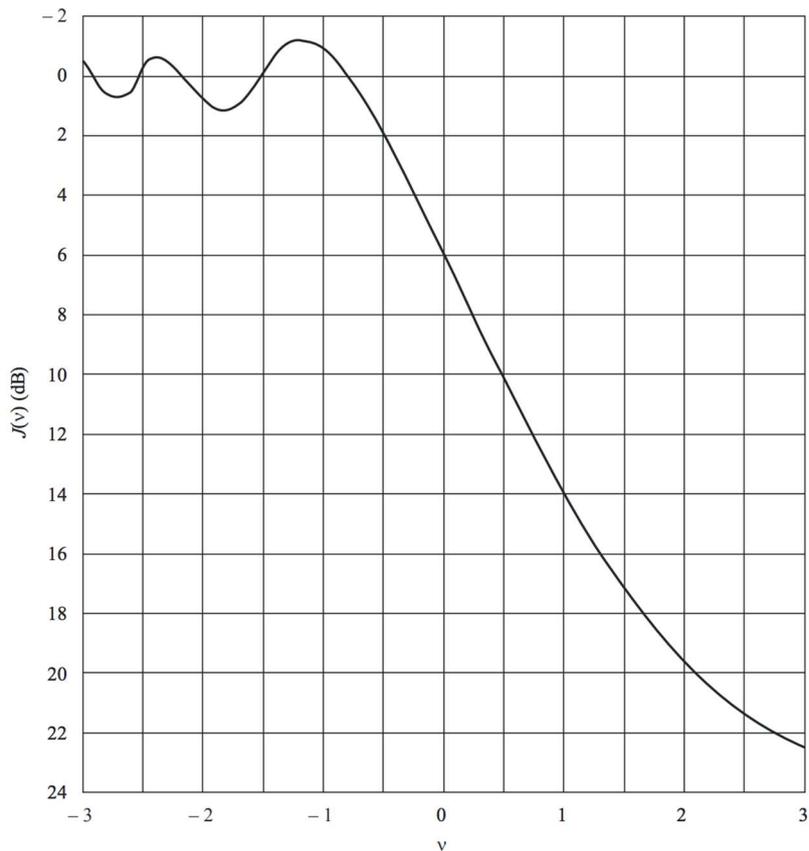


Figura 3-5. Pérdidas obstáculo filo de cuchillo. [9]

3.2.2 Dos obstáculos aislados

El método para obtener las pérdidas que producen dos obstáculos filo de cuchillo, de los cuales no se conoce el radio, consiste en aplicar sucesivamente el procedimiento de difracción explicado en el apartado anterior para un único obstáculo aislado teniendo en cuenta que el primero de ellos actúa como fuente de difracción sobre el segundo.

El primer obstáculo produce unas pérdidas $L_1(dB)$ que dependen de la distancia entre el TX y él (a), la distancia entre los dos obstáculos (b) y de la altura desde la línea que une el TX con la parte alta del obstáculo uno y a línea que va desde el TX hasta el obstáculo dos (h'_1) y es calculada usando el método de un único obstáculo filo de cuchillo. El segundo obstáculo, al igual, dará lugar a unas pérdidas, $L_2(dB)$, que dependen de las distancias b y c , siendo esta última la que existe entre el segundo obstáculo y el RX y de la altura h'_2 .

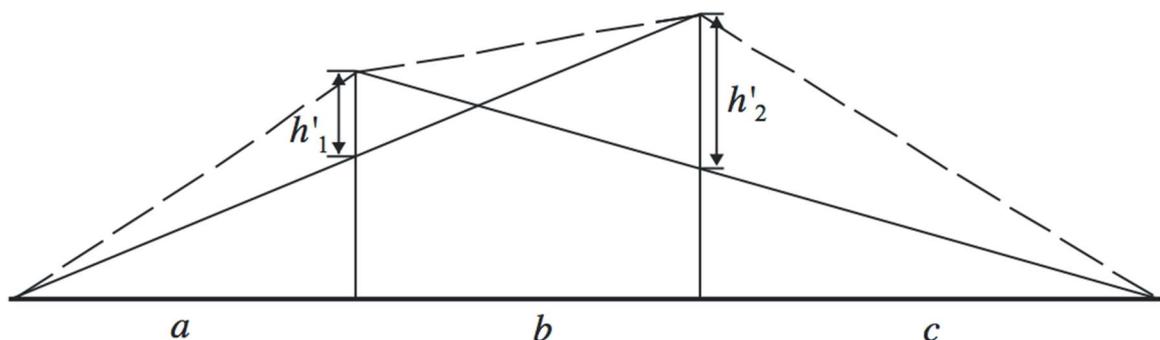


Figura 3-6. Esquema de un radioenlace con dos obstáculos aislados – Filo de cuchillo. [10]

Para poder realizar el cómputo total de las pérdidas que producen ambos obstáculos debe de añadirse un término de corrección $L_c(dB)$ que tiene en cuenta la separación, b , entre ambos y que puede estimarse con la siguiente fórmula siempre que $L_1(dB)$ y $L_2(dB)$ superen los $15dB$:

$$L_c = 10 \log \left[\frac{(a+b)(b+c)}{b(a+b+c)} \right] \quad (3-38)$$

Una vez calculado, se puede concluir con:

$$L = L_1 + L_2 + L_c \quad (3-39)$$

Esto es muy útil cuando ambos obstáculos tienen las mismas pérdidas.

Hay ocasiones en las que uno de los dos predomina, en este caso, la difracción de la primera parte viene definida por las distancias a y $b+c$ y la altura h_1 , y la de la segunda por las distancias b y c y la altura h_2 .

Al igual que antes, el método que se utiliza es aplicar el filo de cuchillo en cada uno de los dos obstáculos, comenzando en este caso por el que de mayor altura o h/r . Este obstáculo será el principal, M donde h será la altura desde el borde del obstáculo hasta el TXRX y r el radio del primer elipsoide de Fresnel con la ecuación (1-2) en cada uno de los puntos donde se encuentran los obstáculos.

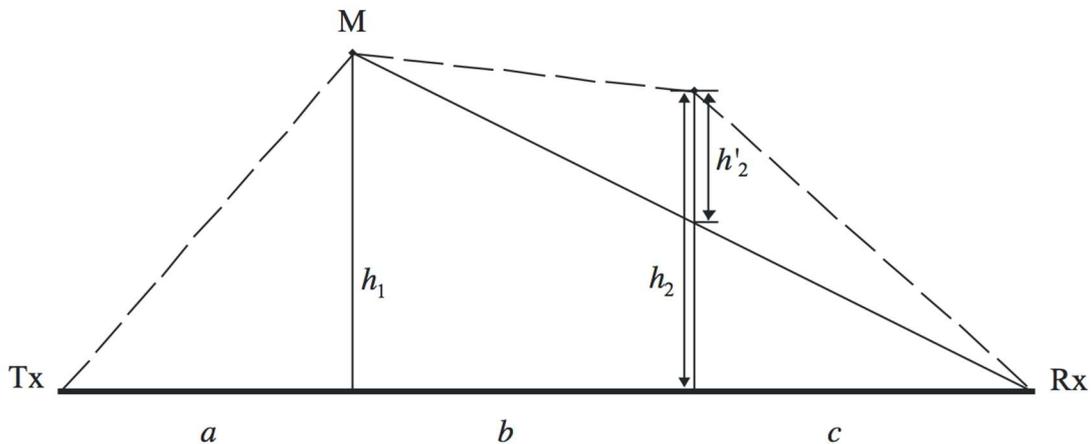


Figura 3-7. Esquema de un radioenlace con dos obstáculos aislados donde uno predomina – Filo de cuchillo. [11]

En este caso, hay que calcular un término de corrección, $T_c(dB)$ para que se pueda tener en cuenta la separación entre los dos bordes, así como la altura de ambos.

Por tanto, las pérdidas totales de difracción vienen dadas por:

$$L = L_1 + L_2 - T_c \quad (3-40)$$

Dónde:

$$T_c = \left[12 - 20 \log \left(\frac{2}{1 - \frac{\alpha}{\pi}} \right) \right] \left(\frac{q}{p} \right)^{2p} \quad (3-41)$$

$$\tan \alpha = \left[\frac{b(a + b + c)}{ac} \right] \quad (3-42)$$

$$p = \sqrt[2]{\frac{2(a + b + c)}{\lambda(b + c)a}} h_1 \quad (3-43)$$

$$q = \sqrt[2]{\frac{2(a + b + c)}{\lambda(a + b)c}} h_2 \quad (3-44)$$

El mismo método puede aplicarse para el caso de obstáculos redondeados.

4 RESOLUCIÓN ANALÍTICA

¡O qué glorioso es el dar! ¡O qué miserable es el recibir! Quanto es mejor el acto de la posesión, tanto es mas noble el dante que el recipiente.

- Fernando de Rojas-

Este apartado concentra la resolución de diversos problemas de radioenlaces que presentan pérdidas por difracción.

Los datos que se van a obtener son realmente importantes para la continuación de este trabajo, pues, más adelante serán comparados con los resultados que obtenidos en el programa.

Hay que tener en cuenta que todos los valores que se van a dar a continuación son aproximaciones y se encuentran redondeados al valor más cercano, lo cual puede producir grandes diferencias con los que se obtendrán en los próximos puntos, sobre todo cuando se trata de decibelios.

4.1 Medio Llano

Todos los problemas que se van a proponer son para radioenlace en un medio llano, en el que ninguno de los obstáculos obstruye el primer elipsoide de Fresnel.

Teóricamente, las coordenadas dónde se encuentran localizados el TX y el RX no son necesarios, pero sí la altura a la que se encuentran ambos, pues haciendo como cero absoluto el lugar donde se encuentra la antena situada a menor elevación, podremos recalcular la altura de la otra antena y así poder realizar el problema lo más cercano a la realidad posible.

4.1.1 Caso 1: Distancia del enlace menor a d_{los} , $h < h_{req}$ y $1 > K > 0.001$

En este caso tendremos como valores conocidos:

- Frecuencia: 500MHz
- a_e : 6370 km
- Distancia que separa ambas antenas: 9.7 km
- Altura del mástil de la antena TX: 1 m
- Altura del mástil de la antena RX: 1 m
- Altura del TX, h_1 : 1 m

- Altura del RX, h_2 : 440 m aproximadamente
- Polarización horizontal
- Para esta frecuencia la longitud de onda $\lambda = 0.6$ m

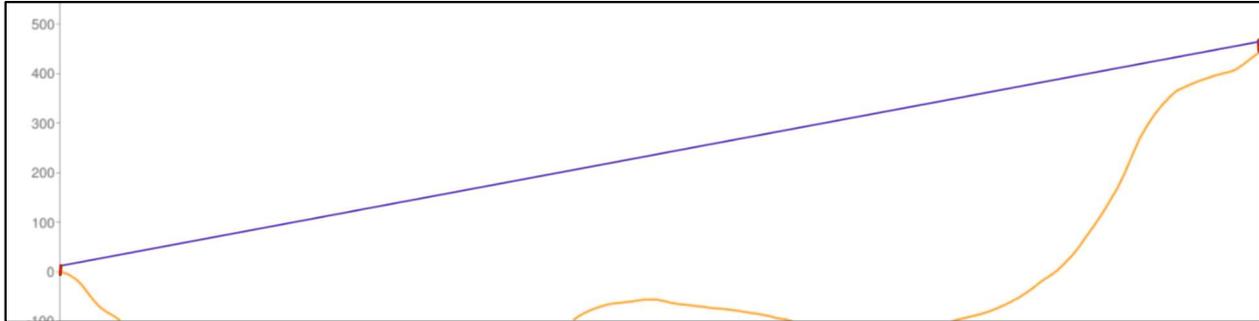


Figura 4-1. Esquema del radioenlace propuesto – Caso 1 Sin Obstáculos.

Comenzamos calculando d_{los} :

$$d_{los} = \sqrt[2]{2 * 6370 * 10^3 (\sqrt[2]{1} + \sqrt[2]{440})} = \mathbf{78439.87 \text{ m}} \quad (4-1)$$

Se verifica que $d > d_{los}$.

Utilizando las gráficas del Anexo I [12] para una frecuencia de 500MHz se obtienen los valores de conductividad y de permitividad relativa, que en este caso son:

$$\varepsilon = 15$$

$$\sigma = 0.015 \text{ S/m}$$

Con estos datos procedemos a calcular h y h_{req} .

$$c = \frac{1 - 440}{1 + 440} = \mathbf{-0.995} \quad (4-2)$$

$$m = \frac{9700^2}{4 * 6370 * 10^3 (1 + 440)} = \mathbf{8.37 * 10^{-3}} \quad (4-3)$$

$$b = 2 \sqrt[2]{\frac{8.37 * 10^{-3} + 1}{3 * 8.37 * 10^{-3}}} \cos \left(\frac{\pi}{3} + \frac{1}{3} \arccos \left(\frac{3 * (-0.995)^2}{2} \sqrt[2]{\frac{3 * 8.37 * 10^{-3}}{(8.37 * 10^{-3} + 1)^3}} \right) \right) = \mathbf{-0.995} \quad (4-4)$$

$$d_1 = \frac{9700}{2} (1 + (-0.995)) = \mathbf{24.25 \text{ m}} \quad (4-5)$$

$$d_2 = 9700 - 24.25 = \mathbf{9675.75 \text{ m}} \quad (4-6)$$

$$h = \frac{\left(1 - \frac{24.15^2}{2 * 6370 * 10^3}\right) * 9675.75 + \left(440 - \frac{9675.75^2}{2 * 6370 * 10^3}\right) * 24.25}{9700} = \mathbf{2.079 \text{ m}} \quad (4-7)$$

$$h_{req} = 0.552 \sqrt[2]{\frac{24.25 * 9675.75 * 0.6}{9700}} = \mathbf{2.1 \text{ m}} \quad (4-8)$$

Se verifica que $h < h_{req}$, por tanto, las pérdidas no son cero y se tiene que continuar con el procedimiento explicado en el apartado “Difracción sobre el horizonte” utilizando un radio efectivo de la tierra modificado de valor:

$$a_{em} = 0.5 \left(\frac{9700}{\sqrt[2]{1} + \sqrt[2]{440}} \right)^2 = \mathbf{97411.26 \text{ m}} \quad (4-9)$$

Como la polarización se dice que es horizontal:

$$K_H = \left(\frac{2\pi * 97411.26}{0.6} \right)^{-1/3} [(15 - 1)^2 + (60 * 0.6 * 0.015)^2]^{-1/4} = \mathbf{2.65 * 10^{-3}} \quad (4-10)$$

El valor de K_H se encuentra entre 1 y 0.001, además se tiene en cuenta que la frecuencia es mayor a 20MHz y que el radioenlace se encuentra en un suelo moderadamente seco, es decir, por tierra, y el valor de $\beta = 1$. Se continua con el procedimiento:

$$X = 9700 * \sqrt[3]{\frac{\pi}{0.6 * (97411.26)^2}} = \mathbf{7.95} \quad (4-11)$$

Para continuar tenemos que analizar este valor de X , que como es mayor de 1.6 se procederá como se indica:

$$F(X) = 11 + 10 \log(7.95) - 17.6 * 7.95 = \mathbf{-119.92 \text{ dB}} \quad (4-12)$$

$$Y(TX) = 2 * 1 * \sqrt[3]{\frac{\pi^2}{0.6^2 * 97411.26}} = \mathbf{0.131} \quad (4-13)$$

$$Y(RX) = 2 * 440 * \sqrt[3]{\frac{\pi^2}{0.6^2 * 97411.26}} = \mathbf{57.669} \quad (4-14)$$

$$B(TX) = \mathbf{0.131} \quad (4-15)$$

$$B(RX) = \mathbf{57.669} \quad (4-16)$$

Como ambas B son menor que 2:

$$G(Y_{TX}) = 20 \log(0.131 + 0.1 * 0.131^3) = \mathbf{-17.639 \text{ dB}} \quad (4-17)$$

$$G(Y_{RX}) = 20 \log(57.669 + 0.1 * 57.669^3) = \mathbf{85.68 \text{ dB}} \quad (4-18)$$

En ninguno de los dos valores anteriores se cumple la condición $G(Y) < 2 + 20 \log K$, por lo que toman los valores que se han calculado en las ecuaciones 3-17 y 3-18 y se puede finalizar con:

$$A_h = -119.92 - 17.639 - 85.68 = \mathbf{-51.879 \text{ dB}} \quad (4-19)$$

Cómo es un valor negativo, **las pérdidas serán cero.**

4.1.2 Caso 2: Distancia del enlace menor a d_{los} y $h > h_{req}$

- Frecuencia: 300MHz
- a_e : 6370 km
- Distancia que separa ambas antenas: 600 m
- Altura del mástil de la antena TX: 4 m
- Altura del mástil de la antena RX: 4 m
- Altura del TX, h_1 : 70 m aproximadamente
- Altura del RX, h_2 : 4 m
- Para esta frecuencia la longitud de onda $\lambda = 1$ m

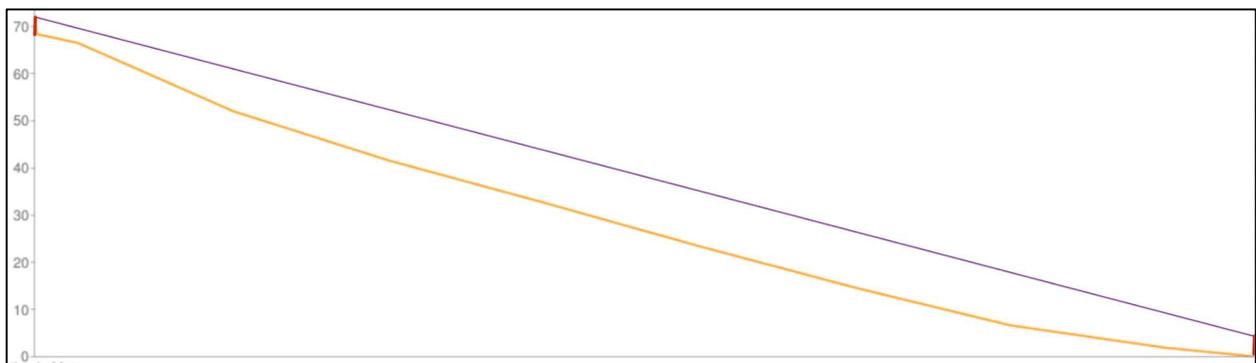


Figura 4-2. Esquema del radioenlace propuesto – Caso 2 Sin Obstáculos.

$$c = \frac{70 - 4}{70 + 4} = \mathbf{0.892} \quad (4-20)$$

$$m = \frac{600^2}{4 * 6370 * 10^3 (70 + 4)} = \mathbf{1.909 * 10^{-4}} \quad (4-21)$$

$$b = 2 \sqrt{\frac{1.909 * 10^{-4} + 1}{3 * 1.909 * 10^{-4}}} \cos \left(\frac{\pi}{3} + \frac{1}{3} \arccos \left(\frac{3 * (0.892)^2}{2} \sqrt{\frac{3 * 1.909 * 10^{-4}}{(1.909 * 10^{-4} + 1)^3}} \right) \right) = \mathbf{0.895} \quad (4-22)$$

$$d_1 = \frac{600}{2} (1 + 0.892) = \mathbf{567.6 \text{ m}} \quad (4-23)$$

$$d_2 = 600 - 567.6 = \mathbf{32.4 \text{ m}} \quad (4-24)$$

$$h = \frac{\left(70 - \frac{567.6^2}{2 * 6370 * 10^3} \right) * 32.4 + \left(4 - \frac{32.4^2}{2 * 6370 * 10^3} \right) * 567.6}{600} = \mathbf{7.48 \text{ m}} \quad (4-25)$$

$$h_{req} = 0.552 \sqrt{\frac{567.6 * 32.4 * 1}{600}} = 3.056 \text{ m} \quad (4-26)$$

Como el valor de h es mayor que h_{req} **no existen pérdidas.**

4.2 Medio con Obstáculos

4.2.1 Caso 1: Un obstáculo “Filo de Cuchillo”

Datos:

- Frecuencia: 2GHz
- a_e : 6370 km
- Distancia que separa ambas antenas: 330 m
- Altura del mástil de la antena TX: 7 m
- Altura del mástil de la antena RX: 7 m
- Altura del TX, h_1 : 7 m
- Altura del RX, h_2 : 7.5 m aproximadamente
- Para esta frecuencia la longitud de onda $\lambda = 0.15\text{ m}$
- Distancia desde el TX al obstáculo: 175 m

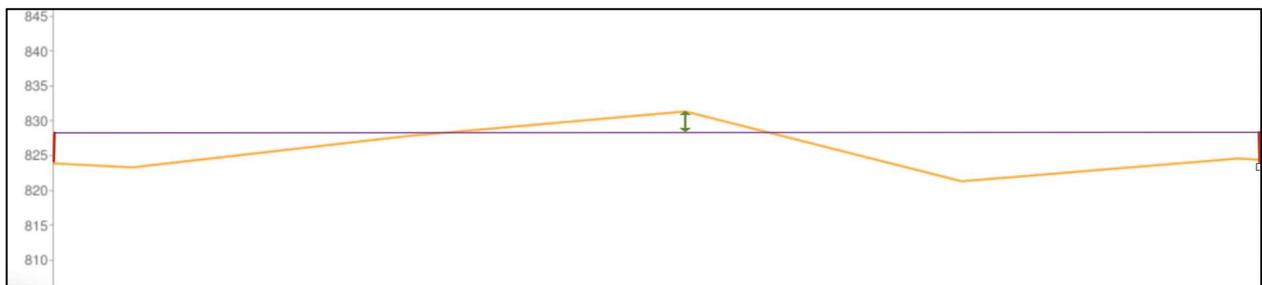


Figura 4-3. Esquema del radioenlace propuesto – Caso 1 Con Obstáculos.

En esta imagen aún no se ha realizado la conversión absoluta de valores para que comiencen en cero.

Se calcula el punto de la línea recta que une la antena transmisora y receptora en el punto en el que se encuentra el obstáculo:

$$y_{175m} = \frac{7.5 - 7}{330} * 175 + 7 = 7.265\text{ m} \quad (4-27)$$

Despejamiento en este punto, teniendo en cuenta que la elevación del obstáculo relativa a la altura de la antena más baja es de 7.45m :

$$h(175m) = 7.45 - 7.265 = 0.185\text{ m} \quad (4-28)$$

Calculando el radio del primer elipsoide de Fresnel en este punto, se verifica que el elemento obstruye sin evitar la conexión:

$$R_1 = \sqrt[2]{\frac{0.15 * 175 * (330 - 175)}{330}} = 3.51\text{ m} \quad (4-29)$$

$$-0.6 \leq \frac{0.185}{3.51} \leq 0.5$$

Para calcular las pérdidas totales que produce hay que calcular v y dependiendo de esta, seguir un método para calcular $J(v)$.

$$v = 0.185 * \sqrt[2]{\frac{2}{0.15} \left(\frac{1}{175} + \frac{1}{155} \right)} = \mathbf{0.0745} \quad (4-30)$$

Como este valor es superior a -0.78 :

$$J(0.0745) = 6.9 + 20 \log \left(\sqrt[2]{(0.0745 - 0.1)^2 + 1} + 0.0745 - 0.1 \right) = \mathbf{6.679 \text{ dB}} \quad (4-31)$$

Se concluye afirmando que este enlace **presenta unas pérdidas del valor de 6.679dB.**

4.2.2 Caso 2: Un obstáculo con radio conocido

Datos:

- Frecuencia: 2GHz
- a_e : 6370 km
- Distancia que separa ambas antenas: 330 m
- Altura del mástil de la antena TX: 7 m
- Altura del mástil de la antena RX: 7 m
- Altura del TX, h_1 : 7 m
- Altura del RX, h_2 : 7.5 m aproximadamente
- Para esta frecuencia la longitud de onda $\lambda = 0.15$ m
- Distancia desde el TX al obstáculo: 175 m
- Radio del obstáculo: 20 m

Como nos encontramos con un problema similar al del apartado anterior, no se va a realizar el procedimiento por el cual se verifica que estamos ante un obstáculo en nuestro radioenlace. Al igual, v y $J(v)$ tampoco variarán, por lo que se tomarán iguales.

La gran diferencia, es que, para el cómputo total de pérdidas, como el radio es conocido, hay que calcular un término que depende del valor de este, $T(m,n)$.

$$m = \frac{20 * \frac{(175 + 155)}{175 * 155}}{\sqrt[3]{\frac{\pi * 20}{0.15}}} = \mathbf{0.0325} \quad (4-32)$$

$$n = \frac{0.185 * \left(\frac{\pi * 20}{0.15}\right)^{2/3}}{20} = \mathbf{0.5178} \quad (4-33)$$

Se haya el producto $m * n = \mathbf{0.0168}$. Como es inferior a 4, se calculará $T(m,n)$ como sigue:

$$T(m,n) = 7.2\sqrt[2]{0.0325} - (2 - 12.5 * 0.5178) * 0.0325 + 3.6 * 0.0325^{3/2} - 0.8 * 0.0325^2 \quad (4-34)$$

$$= \mathbf{1.464}$$

$$L = 1.464 + 6.679 = \mathbf{8.14 \text{ dB}}$$

Las pérdidas totales que presenta este enlace con radio conocido son, por tanto, **8.14 dB**

4.2.3 Caso 3: Dos obstáculos aislados “Filo de Cuchillo”

Datos que se conocen:

- Frecuencia: 1GHz
- a_e : 6370 km
- Distancia que separa ambas antenas: 36 km
- Altura del mástil de la antena TX: 5 m
- Altura del mástil de la antena RX: 3 m
- Altura del TX, h_1 : 270 m aproximadamente
- Altura del RX, h_2 : 3 m
- Para esta frecuencia la longitud de onda $\lambda = 0.3$ m
- Distancia desde el TX al obstáculo 1: 8720 m
- Distancia desde el obstáculo 1 al obstáculo 2: 14.5 km

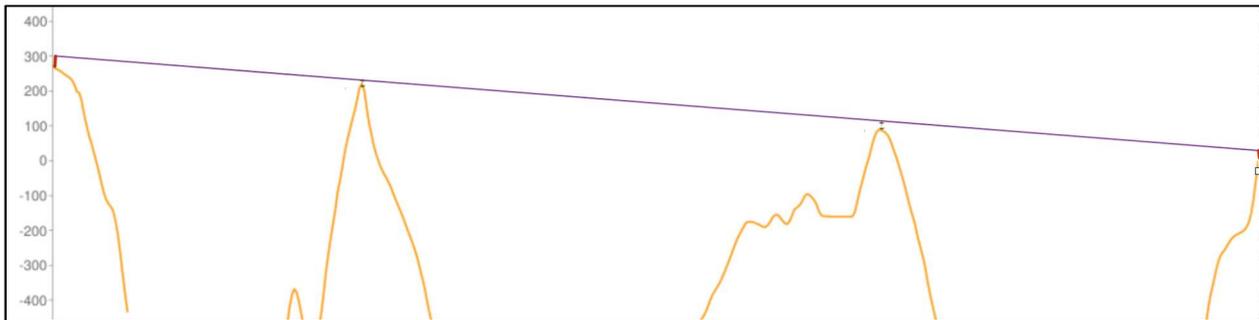


Figura 4-4. Esquema del radioenlace propuesto – Caso 3 Con Obstáculos.

En primer lugar, hay que seguir el procedimiento utilizado en los dos puntos anteriores para calcular las pérdidas producidas por el primer obstáculo.

$$y_{8720m} = \frac{3 - 270}{36000} * 8720 + 270 = \mathbf{205.33\ m} \quad (4-35)$$

La elevación del primer obstáculo relativa a la altura de la antena más baja es de 221m:

$$h(8720m) = 221 - 205.33 = \mathbf{15.67\ m} \quad (4-36)$$

Calculando el radio del primer elipsoide de Fresnel en este punto, se verifica que el elemento obstruye sin evitar la conexión:

$$R_1 = \sqrt{\frac{0.3 * 8720 * (36000 - 8720)}{36000}} = \mathbf{44.52\ m} \quad (4-37)$$

$$-0.6 \leq \frac{15.67}{44.52} \leq 0.5$$

Se calcula v :

$$v = 15.67 * \sqrt[2]{\frac{2}{0.3} \left(\frac{1}{8720} + \frac{1}{27280} \right)} = \mathbf{0.498} \quad (4-38)$$

Como es superior a -0.78 :

$$J(0.498) = 6.9 + 20 \log \left(\sqrt[2]{(0.498 - 0.1)^2 + 1} + 0.498 - 0.1 \right) = \mathbf{10.27 \text{ dB}} \quad (4-39)$$

Para calcular las pérdidas producidas por el segundo obstáculo seguimos este mismo procedimiento, pero, hay que calcular una nueva altura relativa a la línea recta que une el obstáculo con la antena RX y trabajar con este valor. Se tiene en cuenta también que la distancia del obstáculo 1 al RX es de 27280m y que la altura del obstáculo 2 es de 89m relativos.

$$y_{15km} = \frac{3 - 15.67}{27280} * 15000 + 15.67 = \mathbf{8.7 \text{ m}} \quad (4-40)$$

El nuevo despejamiento es:

$$h'_2(15km) = 89 - 8.7 = \mathbf{80.3 \text{ m}} \quad (4-41)$$

Con estos datos se puede obtener v y $J(v)$. Se calcula la distancia del obstáculo 2 al RX con los datos que tenemos y obtenemos que es m, con esto, se procede:

$$v = 80.3 * \sqrt[2]{\frac{2}{0.15} \left(\frac{1}{15000} + \frac{1}{12280} \right)} = \mathbf{3.56} \quad (4-42)$$

Superior a -0.78 :

$$J(3.56) = 6.9 + 20 \log \left(\sqrt[2]{(3.56 - 0.1)^2 + 1} + 3.56 - 0.1 \right) = \mathbf{23.878 \text{ dB}} \quad (4-43)$$

En estos casos en los que se presentan dos obstáculos aislados (distancia entre ellos de más de 250m) hay que calcular un término de corrección, $T_c(\text{dB})$ para que se pueda tener en cuenta la separación entre los dos bordes, así como la altura de ambos.

Por tanto, las pérdidas totales de difracción vienen dadas por:

$$L = L_1 + L_2 - T_c \quad (4-44)$$

Dónde:

$$p = \sqrt[2]{\frac{2(8720 + 15000 + 12280)}{0.3(15000 + 12280) * 8720}} * 15.67 = \mathbf{0.45} \quad (4-45)$$

$$q = \sqrt[2]{\frac{2(8720 + 15000 + 12280)}{0.3(8720 + 15000)12280}} * 15.78 = \mathbf{0.026} \quad (4-46)$$

$$\alpha = \text{atan} \left[\frac{15000(8720 + 15000 + 12280)}{8720 * 12280} \right] = \mathbf{1.12} \quad (4-47)$$

$$T_c = \left[12 - 20 \log \left(\frac{2}{1 - \frac{1.12}{\pi}} \right) \right] \left(\frac{0.026}{0.45} \right)^{2*0.45} = \mathbf{0.165} \quad (4-48)$$

$$L = 10.27 + 23.878 - 0.165 = \mathbf{33.983 \text{ dB}} \quad (4-49)$$

Se concluye con un **valor de pérdidas totales igual a 33.983 dB** en este enlace con dos obstáculos aislados.

4.2.4 Caso 4: Dos obstáculos aislados con radios conocidos

Usando los mismos datos que el caso anterior, obtendremos el mismo valor de pérdidas, pero, como ahora los radios son conocidos habrá que tener en cuenta el factor $T(m,n)$.

Datos:

- Radio del primer obstáculo: 15 m
- Radio del segundo obstáculo: 25 m

Para el primero tendremos un valor de $T(m,n)$ de **2.056**

$$m = \frac{15 * \frac{(8720 + 27280)}{8720 * 27280}}{\sqrt[3]{\frac{\pi * 15}{0.3}}} = 4.2 * 10^{-3} \quad (4-50)$$

$$n = \frac{15.67 * \left(\frac{\pi * 15}{0.3}\right)^{2/3}}{15} = 30.41 \quad (4-51)$$

Se haya el producto $m * n = 0.128$. Como es inferior a 4, se calculará $T(m,n)$ como sigue:

$$T(m,n) = 7.2^2 \sqrt[4]{4.2 * 10^{-3}} - (2 - 12.5 * 30.41) * 4.2 * 10^{-3} + 3.6 * 4.2 * 10^{-3^{3/2}} - 0.8 * 4.2 * 10^{-3^2} = 2.056 \quad (4-52)$$

$$L_1 = 2.056 + 10.27 = 12.326 \text{ dB}$$

Para el segundo:

$$m = \frac{25 * \frac{(15000 + 12280)}{15000 * 12280}}{\sqrt[3]{\frac{\pi * 25}{0.3}}} = 5.788 * 10^{-4} \quad (4-53)$$

$$n = \frac{80.3 * \left(\frac{\pi * 25}{0.3}\right)^{2/3}}{25} = 131.45 \quad (4-54)$$

Se haya el producto $m * n = 0.076$. Como es inferior a 4, se calculará $T(m,n)$ como sigue:

$$T(m,n) = 7.2^2 \sqrt[4]{5.788 * 10^{-4}} - (2 - 12.5 * 131.45) * 5.788 * 10^{-4} + 3.6 * 5.788 * 10^{-4^{3/2}} - 0.8 * 5.788 * 10^{-4^2} = 1.12 \quad (4-55)$$

$$L_1 = 1.12 + 23.878 = 24.998 \text{ dB}$$

(4-56)

Finalmente se pueden obtener las pérdidas totales:

$$L = 12.326 + 24.998 - 0.165 = \mathbf{37.159\ dB}$$

Las pérdidas totales que presenta este enlace con radio conocido son, por tanto, 37.159dB

4.2.5 Caso 5: Obstáculo que hace inviable el radioenlace

Datos que se conocen:

- Frecuencia: 2GHz
- a_e : 6370 km
- Distancia que separa ambas antenas: 2 km
- Altura del mástil de la antena TX: 10 m
- Altura del mástil de la antena RX: 10 m
- Altura del TX, h_1 : 60 m aproximadamente
- Altura del RX, h_2 : 10 m
- Para esta frecuencia la longitud de onda $\lambda = 0.15 \text{ m}$
- Distancia desde el TX al obstáculo: 1670 m

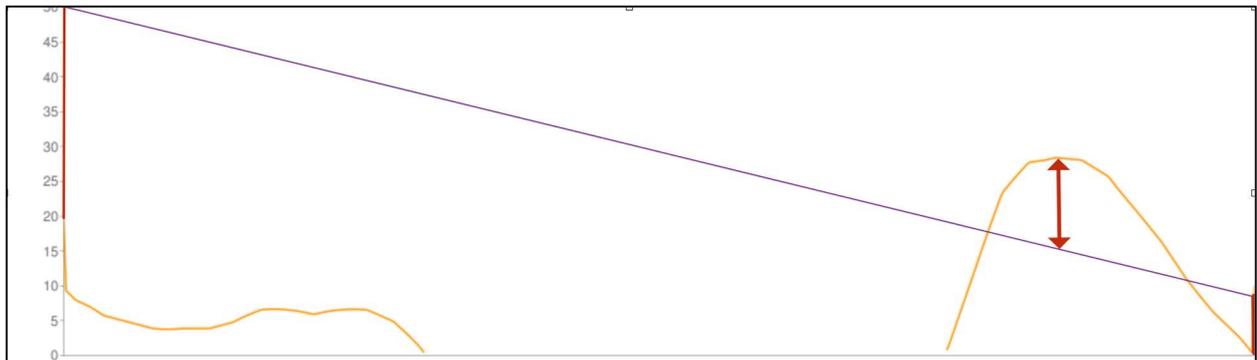


Figura 4-5. Esquema del radioenlace propuesto – Caso 5 Con Obstáculos.

Calculamos la obstrucción que presenta la primera zona de Fresnel:

$$y_{1670m} = \frac{10 - 60}{2000} * 1670 + 60 = \mathbf{18.25 \text{ m}} \quad (4-57)$$

$$h(1670m) = 64 - 18.25 = \mathbf{45.75 \text{ m}} \quad (4-58)$$

Calculando el radio del primer elipsoide de Fresnel en este punto:

$$R_1 = \sqrt{\frac{0.15 * 1670 * (2000 - 1670)}{2000}} = \mathbf{6.429 \text{ m}} \quad (4-59)$$

$$\frac{45.75}{6.429} > 0.5$$

Obstruye más del 60% de la primera zona de Fresnel en ese punto. **Enlace inviable.**

5 PROGRAMA EN PYTHON

Lo esencial es invisible a los ojos.

- Antoine De Saint-Exupéry -

Python es un lenguaje de programación que se caracteriza por presentar una sintaxis simple y limpia. Posee gran cantidad de librerías, entre ellas algunas muy completas y robustas con ejecuciones matemáticas y operaciones simplificadas, lo que lo hace ideal para ser el elegido a la hora de realizar un código en el que se puedan implementar las fórmulas presentadas con anterioridad y obtener resultados con gran aproximación y precisión rozando el nivel de otros grandes como Matlab.

5.1 Programa principal: TFG_Principal.py

Este fichero incluye el programa principal desde el cual se llama al resto de funciones [13] que se explican más adelante dónde se calculan los parámetros intermedios.

Para comenzar, tenemos que introducir por teclado algunos valores que son indispensables para que el programa pueda comenzar a funcionar, como son la frecuencia, la altura de los mástiles de las antenas, el radio efectivo de la tierra, y las localizaciones en las cuales se quiere colocar. Se calcula la longitud de onda, la distancia total y se inicializa muestras, con lo que se llama a la primera función necesaria de la que obtendremos tres tablas que incluyen elevación, latitud y longitud de esos puntos donde se está midiendo el radioenlace.

Como el enlace se encuentra en un terreno real, ambas antenas no se encontrarán a la misma altura, por lo que hay que hacer un cero absoluto en aquella que tiene menor elevación y tras esto, restarles ese valor a todos los puntos del medio, quedando todo sobre un valor de elevación de referencia.

Para continuar, se llama a la función que nos devuelve el tipo de terreno en el que nos encontramos para así poder continuar con el procedimiento que corresponda. En el caso de ser [14] un medio llano con frecuencia menor a 10MHz o el radio efectivo es 0, tendrá que utilizarse el programa de la ITU-R GRWAVE, en el caso de ser mayor y un radio efectivo superior, se llamará a la función que nos devuelve las pérdidas correspondientes en ese enlace y que se imprime por pantalla dando el resultado final en decibelios.

Por otro lado, si el medio se prevé que pueda tener algún obstáculo, se llama la función que nos dice el número exacto de estos, si es uno o dos, tendrán que calcularse las pérdidas exactas utilizando los métodos que se explicaron en apartados anteriores. A excepción del caso en el que el enlace presente dos obstáculos, pero la distancia que los separa es inferior a 250m, para el cuál

no se ha implementado el método en este código y no se puede suponer que son dos obstáculos aislados.

Si tiene más de dos, imprimirá por pantalla el número total de estos. También, se incluyen una serie de errores que se pueden dar y que obligan a la finalización inmediata del programa.

Las funciones son parte básica e imprescindible a la hora de programar pues, hacen que el código este estructurado y modulado, además consigue que sea más fácil la detección de posibles errores.

En este se han programado 14 funciones divididas en distintos ficheros según cuando son utilizadas en el programa principal. A continuación, se explican cada una de estas en unos apartados titulados como el fichero en el que se pueden encontrar [15].

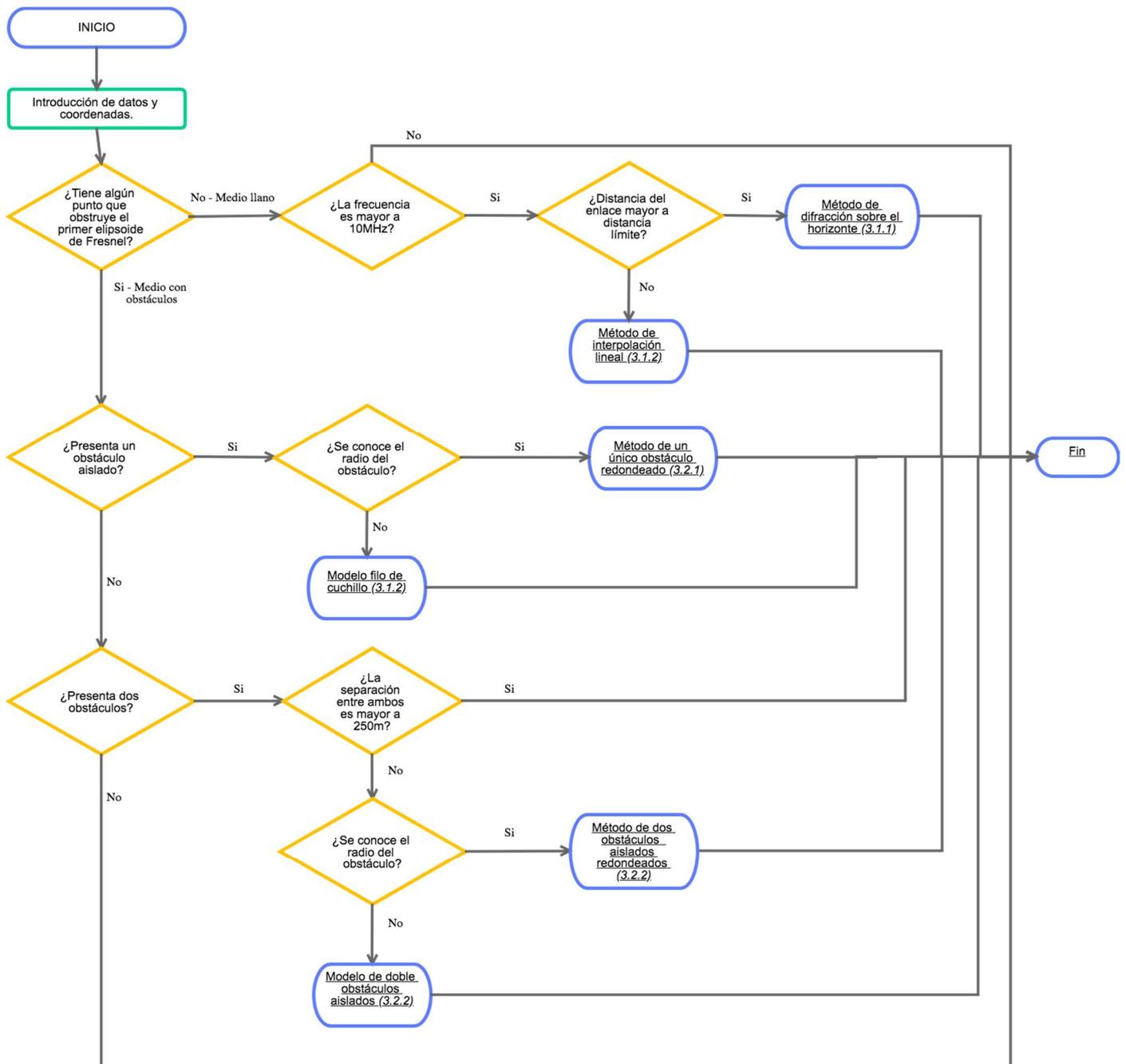


Figura 5-1. Diagrama de flujo del programa.

5.2 Funciones utilizadas: Terreno.py

Este fichero incluye aquellas funciones que se encargan de analizar el perfil y el terreno en el cual estamos trabajando para caracterizarlo según los tipos que hemos nombrado a lo largo del trabajo.

5.2.1 perfil_terreno(latTX, lonTX, latRX, lonRX, muestras)

Esta función es imprescindible para el correcto funcionamiento del resto del programa. Comienza introduciendo como parámetros la latitud y longitud de la antena transmisora y receptora, así como el número de muestras que se van a tomar del perfil. Tras esto, la guarda en otras variables con el formato necesario para la función *getElevation(enlace,muestras)* que devuelve una tabla con la elevación del terreno, la longitud y la latitud de esos puntos.

5.2.2 terrain_type(dis, elevacion, wave_len, a_e, h_TX, h_RX, latitud, longitud, muestras)

La función *terrain_type* recibe por parámetros la distancia total del radioenlace, la tabla de elevación, la longitud de onda a la frecuencia de trabajo, el radio efectivo de la tierra, las alturas de ambas antenas, las tablas de latitud, longitud y el número de muestras.

Con estos parámetros se van a calcular las distancias a las antenas transmisoras y receptoras y el punto de la recta que las une a esa latitud y longitud. Este último punto calculado se compara con la elevación que hay en las coordenadas de estudio en ese instante.

Tras esto se obtiene el radio del primer elipsoide de Fresnel y según la obstrucción que presente se tendrá un medio llano o con obstáculos. En el caso en que uno de los puntos ya obstruya, la función finalizará devolviendo que el medio tiene al menos, un posible obstáculo.

5.3 Funciones utilizadas: apiTFG.py

Las funciones que se encontrarán en este documento harán uso de la API de Google que usando Maps nos devolverá datos imprescindibles para el análisis del radioenlace, así como funciones de cálculo de distancias.

5.3.1 getElevation(path, samples, sensor="false", **elvt_n_args)

Esta función implementa la API de Google Maps [16] que con la introducción de dos posiciones te devuelve el perfil del enlace que une ambas localizaciones. [17]

En este caso se analizan 500 puntos y se devuelve la elevación, la latitud y la longitud de estos. Esta API y su implementación se explican con más detalle en el Anexo III.

5.3.2 GetDistance(lat1, lon1, lat2, lon2, R)

Introduciéndole dos posiciones (latitudes y longitudes) así como el radio efectivo de la tierra y haciendo uso de determinadas fórmulas matemáticas se calcula la distancia que hay entre esos dos puntos.

5.4 Funciones utilizadas: TerrenoLlano.py

El fichero TerrenoLlano.py introduce aquellas funciones que realizan los cálculos necesarios para obtener el valor de las pérdidas totales que se producen en este tipo de medio.

5.4.1 medio_llano(a_e, h_TX, h_RX, dis, wave_len, freq)

Una vez comprobado que no se encuentra ningún obstáculo que obstruya la primera zona de Fresnel, en esta función se calcula la distancia límite con la que se comparará la distancia del enlace que estamos analizando.

Si es mayor, se calcularán las pérdidas utilizando un método (calculando K y la fuerza de difracción del campo respecto a la del espacio libre) y si es menor, con la altura del radio de curvatura del rayo.

Devuelve al programa principal las pérdidas totales que se producen en este tipo de medios.

5.4.2 admitancia_superficie(wave_len, a_e)

La admitancia en superficie es un término que además de la longitud de onda y del radio efectivo de la tierra, depende de otros que se van introduciendo por teclado cuando el programa va a necesitarlos, como es, entre otros, la polarización. Por pantalla encontraremos una frase donde se nos indica que hay que introducir “h” o “v”, aunque también se está tomando que estas puedan introducirse en mayúsculas.

Tras la introducción de este valor indispensable para continuar, se realizan las operaciones necesarias para devolver K . Si no se introduce un valor correcto se devuelve “Error”, que será analizado y q hará que el programa finalice.

Para la resolución del problema hemos aproximado la solución solo atendiendo al primer término de la serie, dando por hecho que el error con el valor real de tenerlos todos en cuenta, no va a superar los $2dB$.

5.4.3 fuerza_difraccion_campo(K, dis, wave_len, a_e, freq)

Una vez que tenemos el parámetro K podemos resolver la ecuación 2-4 y obtener la fuerza de difracción del campo. En el caso de que K fuera devuelto como “Error”, este nuevo valor también lo será y el programa finalizará.

Para cualquiera de los otros tres casos, se analizan y se realizan las operaciones correspondientes a cada uno de ellos para que esta función devuelva el valor correcto de la fuerza de difracción del campo respecto a la del espacio libre y medida en dB .

En el caso más común, donde $K > 0.001$ y $K < 1$, se pedirá que sea introducido por teclado si el radioenlace es sobre tierra o mar. Este parámetro es indispensable para calcular β , que se necesita para el resto de cálculos, y que se calcula en la función *calc_beta* que se explica en el próximo punto.

También es necesaria la función *calc_G*, dónde se obtendrá el valor de la ganancia de altura, tanto de la antena TX como de la RX. Los valores que se reciben de las llamadas a esta función también son parte indispensable de la ecuación 2-4, y que por tanto sin ellos no se podría calcular el valor final. Esta función será explicada también más adelante.

5.4.4 `calc_beta(K, tierra_o_mar, freq)`

Esta función calcula β en función de K , de la frecuencia de trabajo y del terreno en el cual está el radioenlace. Se siguen las pautas que se explicaron en “*Difracción por el horizonte*”- “*Cálculo numérico*”.

5.4.5 `calc_G(Y, beta, K)`

Llamando a esta función se obtendrá el valor de la ganancia de altura en función de Y (TX o RX), del parámetro β obtenido de `calc_beta` y de K .

Se tiene en cuenta la excepción explicada en la cual si la ganancia de altura era menor que el valor correspondiente al dado por $2 + 20 \log K$, G tomara este valor.

Esta función será llamada dos veces para calcular la ganancia de la antena TX y de la RX.

5.4.6 `h_curvatura_rayo(dis, h_TX, h_RX, a_e, wave_len)`

La función de este código es el cálculo de la altura mínima desde el rayo que viaja entre la antena TX y RX y la curvatura de la tierra.

Incluye las fórmulas intermedias necesarias para este cálculo, como son las de la distancia desde el TX hasta el punto donde la altura desde la curvatura de la tierra hasta el rayo es más pequeña, y desde esta al RX.

Con estos valores se llama a la función `pérdidas`, que se explica a continuación y cuyo resultado es lo que será devuelto.

5.4.7 `pérdidas(h, d_TX, d_RX, wave_len, dis)`

Esta función es llamada desde `h_curvatura_rayo` y sirve para calcular las pérdidas totales que se producen en el trayecto a causa de la difracción.

Se comienza calculando la mínima altura para la cual no existen pérdidas (h_{req}) y si h es mayor que este valor, no existirán pérdidas.

En cualquier otro caso hay que aplicar el procedimiento explicado en “*Procedimiento de interpolación lineal*”.

5.5 Funciones utilizadas: `TerrenoObstaculos.py`

Al igual que el fichero anterior, este incluye todas las funciones para saber el número exacto de obstáculos que se presentan en el terreno, así como, todas aquellas que calculan las pérdidas producidas por estos.

5.5.1 `numero_obstaculos(dis, latitud, longitud, elevacion, wave_len, a_e, muestras)`

Esta función calcula el número de obstáculos que se encuentran en el perfil, así como la distancia al TX y al RX desde estos, su altura, su latitud, longitud o el despejamiento frente a la línea recta que une ambas antenas.

Introduce como parámetros la distancia total del enlace, las tablas de longitud, latitud y elevación, la longitud de onda, el radio efectivo de la tierra y el número de muestras que se están tomando.

Lo primero que se realiza es guardar en una tabla [18] llamada “posibles[]” aquellos valores que son “picos”, es decir, que su altura es mayor a la del siguiente punto y mayor a la del anterior. Una vez rellena, se realiza un bucle [19] donde se va a ir recorriendo.

Una vez dentro, se analiza cual de esos valores obstruyen el radioenlace haciéndolo inviable, cuál de ellos no obstruyen y cuál se encuentran en la zona en la cual si es viable el radioenlace, pero obstaculizan y producen pérdidas. Es en este último caso en el cual se incrementa en 1 la variable que indica el número de obstáculos reales, además se guardan todos los datos necesarios para ser devueltos y que han sido nombrados con anterioridad.

5.5.2 unico_obs(wave_len, dis, d1, d2, despejamiento)

Esta función es llamada cuando el número de obstáculos es uno. Se encarga de calcular las pérdidas que produce este, tanto si es “filo de cuchillo”, como si el radio es conocido. Si esto último es así, habrá que introducir por teclado el valor de este.

Calcula ν (valor que depende de la geometría del obstáculo y de las distancias a las antenas) para poder introducirlo en la llamada a la función que calcula $J(\nu)$ y también llama a $T(m,n)$ en caso de que el radio sea conocido e introducido por parámetros.

5.5.3 calc_Jv(ν)

Según el valor de ν , introducido por parámetros, se calculará de una forma u otra $J(\nu)$, tal y como se explicó en el punto *Propagación por difracción - Medio con obstáculos*. En el caso de que ν sea inferior a -0.78 , es imprescindible calcular la parte real e imaginaria de las integrales complejas de Fresnel. Como la realización de integrales en Python [20] era complejo, se adjuntado un archivo llamado *Cv_Sv.py* en el cual se incluye un diccionario que contiene parámetros de ν separados 0.01 [21] relacionados con su $C(\nu)$ y $S(\nu)$ que es lo que buscamos. Esta función, por tanto, devuelve el valor de $J(\nu)$ calculado.

5.5.4 calc_Tmn(r, wave_len,d1,d2,h)

Si el radio del obstáculo es conocido, habrá que calcular los parámetros m y n indispensables para obtener el valor de $T(m,n)$ que es lo que se calcula en esta función y el valor que es devuelto.

5.5.5 dos_obs(wave_len,dis,d1,d2,despejamiento,elev,lat,lon,a_e,h_RX)

Cuando el enlace se encuentra obstruido por dos obstáculos, se realizarán los mismos cálculos que en el caso de uno, con algunas modificaciones.

Se calculan ν y $J(\nu)$ para el obstáculo uno de la misma forma que se ha realizado con anterioridad y para el segundo se comienza calculando la distancia de este al obstáculo uno [18]. Tras esto, se traza una línea recta que une el primer obstáculo con el receptor y se obtiene el

punto que corresponde al lugar donde se encuentra el que estamos analizando ahora. Tras esto, se obtiene una nueva altura o h'_2 , y con esta el despejamiento.

Con todos los nuevos datos, se procede para calcular v y $J(v)$ de este obstáculo y con ello se llama la función que nos devolverá el parámetro de corrección T_c que hay que sustraer a la suma de las pérdidas producidas por cada obstáculo.

En el caso de que sean conocidos los radios de ambos, se introducirán por parámetros y se calcularán $T(m,n)_1$ y $T(m,n)_2$ que se sumarán a las $J(v)$ calculadas e igualmente se restará el parámetro de corrección T_c .

5.5.6 calc_Tc(wave_len,d1,d2,dis_obs,despejamiento,h_nueva)

El término de corrección es un valor que hay que calcular cuando el radioenlace presenta dos obstáculos aislados en los que uno predomina frente al otro.

Se calcula siguiendo lo explicado en *Propagación por difracción - Medio con obstáculos*, tras tener el valor de p, q y alpha.

6 COMPARACIÓN DE RESULTADOS

Este es mi sitio. Mi terreno. Campo de aterrizaje de mis ansias. Cielo al revés. Es mi sitio y no lo cambio por ninguno. Caí. No me arrepiento.

- Blas de Otero -

Una vez finalizado el código, se han realizado pruebas sobre diferentes terrenos que cumplen aproximadamente con las mismas características que se propusieron en el apartado 3 *Resolución Analítica*. El fin de este apartado es verificar el correcto funcionamiento del programa realizado en Python.

Las coordenadas del terreno y la vista preliminar del perfil de elevación se han obtenido con el programa de Google llamado “Google Earth” [19] creando rutas que se adaptan a cada uno de los casos y así conseguir que el programa sea lo más realista posible.

Para el caso de medio llano en el que la distancia del enlace es mayor a d_{los} no se ha encontrado un ejemplo válido, pero se ha comprobado mediante un ejemplo ficticio que el código funciona de forma correcta.

6.1 Medio Llano

6.1.1 Caso 1: Distancia del enlace menor a d_{los} y $1 > K > 0.001$

El perfil utilizado para probar el programa en este primer caso, es el que se adjunta en la imagen de la página siguiente, de donde se puede obtener las coordenadas de las antenas, que son:

TX:

Latitud: 57.50632

Longitud: -5.773517

RX:

Latitud: 57.441848

Longitud: -5.663915

Visualmente, se puede comprobar que, en este enlace, no va a haber ningún obstáculo que obstruya nuestro enlace, es por esto, que se ha tomado como ejemplo.

Se va a introducir cuando se pida, los mismos datos de los que se partió en la resolución analítica.

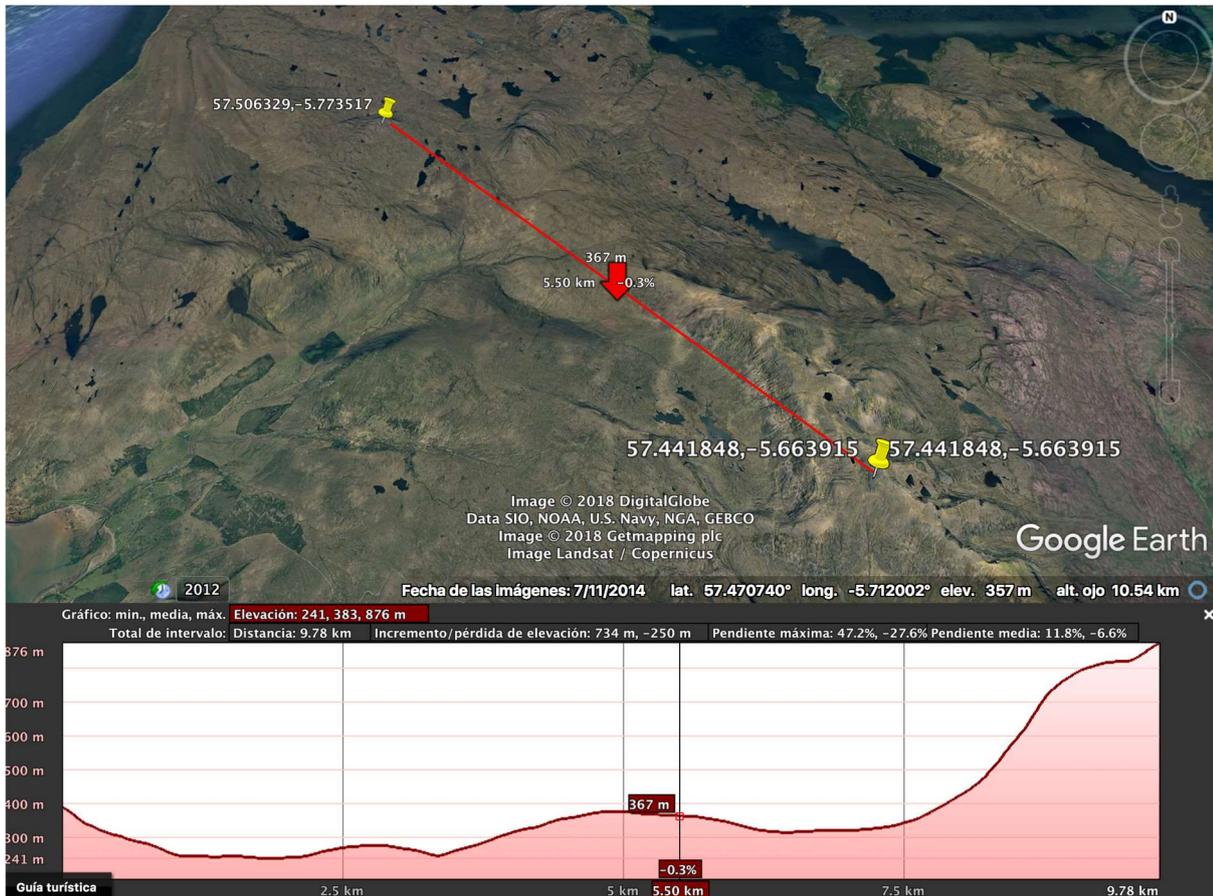


Figura 6-1. Enlace y perfil elevación Google Earth.

Simulación:

```

Bienvenido al simulador de radioenlaces con obstáculos. Se necesitan los siguientes datos para el estudio:
Frecuencia en Hz: 500e6
Altura de la antena TX (mástil) en m: 1
Altura de la antena RX (mástil) en m: 1
Radio efectivo de la tierra en km: 6370
Introduce la latitud de la antena TX en formato decimal: 57.50632
Introduce la longitud de la antena TX en formato decimal: -5.773517
Introduce la latitud de la antena RX en formato decimal: 57.441848
Introduce la longitud de la antena RX en formato decimal: -5.663915
El radio efectivo de la tierra modificado es de: 96.1691244566 km
Polarización de la antena TX(introducir h, para indicar horizontal, o v, para vertical): h
Permitividad relativa: 15
Conductividad en S/m: 0.015
¿La distancia que separa las antenas es por tierra o mar? Introduce t o w: t
El parametro beta que depende del tipo de terreno toma el valor: 1
F(X) en dB: -121.341224046
La ganancia de altura de la antena transmisora,G(Y_1), toma el valor en dB: -17.597953292
La ganancia de altura de la antena receptora,G(Y_2), toma el valor en dB: 116.929613662
Las pérdidas totales producidas por la difracción en este radioenlace que no se encuentra obstruido por ningun obstáculo, son en dB: 0

***Repl Closed***

```

Figura 6-2. Ejecución Enlace 6-1.

Tras la ejecución del código se verifica que con los datos introducidos y con el perfil de la figura 5-1 los resultados que se obtienen son similares a los obtenidos en el apartado 4.1.1. Las variaciones en los decimales se producen como consecuencia de un redondeo evidente en la resolución analítica y que no se realiza en Python, por tanto, las pérdidas que se obtienen son lo más cercanas posibles a la realidad.

6.1.2 Caso 2: Distancia del enlace menor a d_{los} y $h > h_{req}$

Para este, el perfil que se ha tomado de ejemplo en Google Earth es el que tiene las coordenadas de antenas:

TX:

Latitud: 39.2416

Longitud: -6.41

RX:

Latitud: 39.247

Longitud: -6.41

El enlace que se ha tomado presenta a simple vista un medio totalmente llano sin obstáculos que puedan interferir en la comunicación.

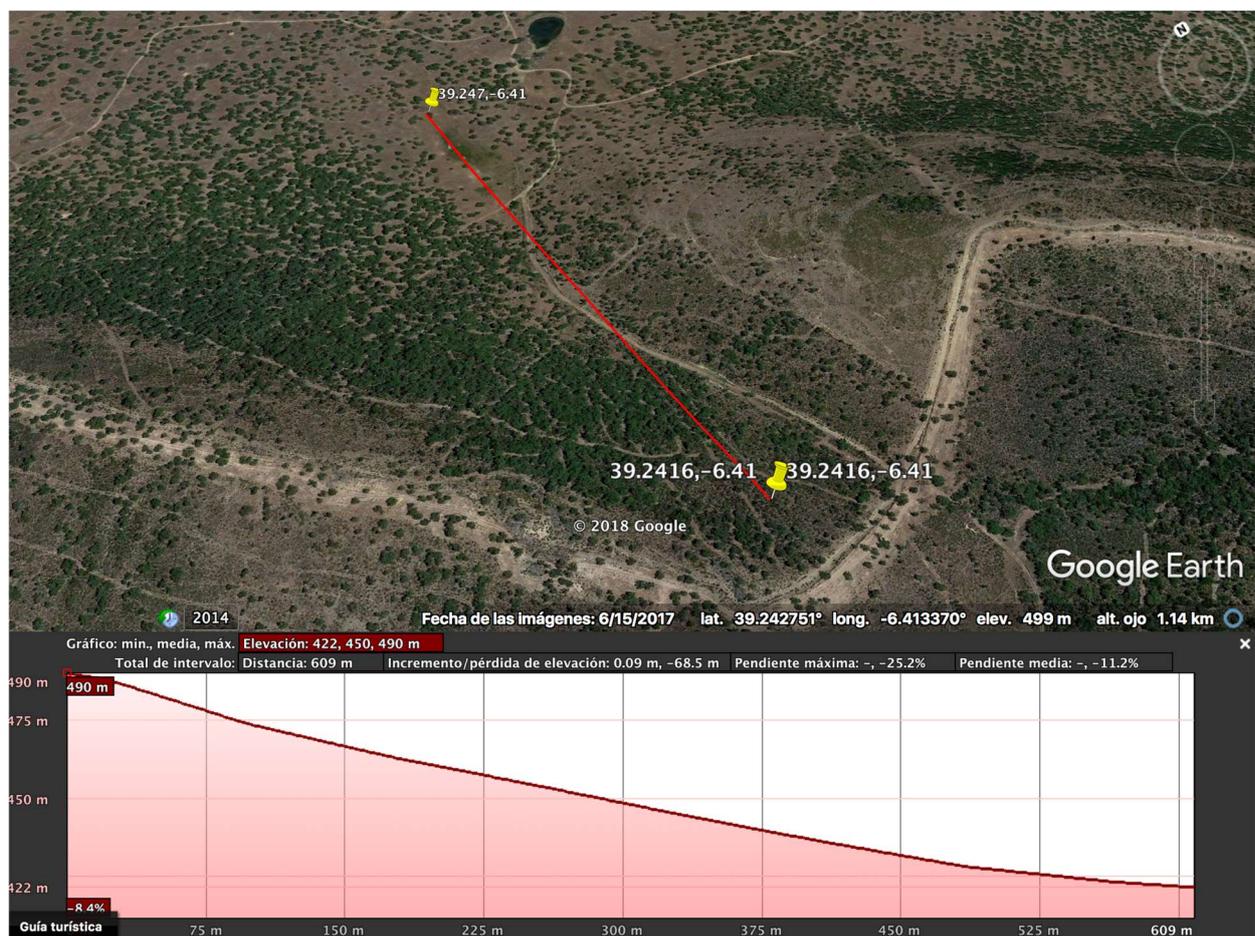


Figura 6-3. Enlace y perfil elevación Google Earth.

Ejecutando el código:

```
Bienvenido al simulador de radioenlaces con obstaculos. Se necesitan los siguientes datos para el estudio:
Frecuencia en Hz: 300e6
Altura de la antena TX (mástil) en m: 4
Altura de la antena RX (mástil) en m: 4
Radio efectivo de la tierra(a_e) en km: 6370
Introduce la latitud de la antena TX en formato decimal: 39.2416
Introduce la longitud de la antena TX en formato decimal: -6.41
Introduce la latitud de la antena RX en formato decimal: 39.247
Introduce la longitud de la antena RX en formato decimal: -6.41
Las pérdidas totales por difracción son en dB: 0

***Repl Closed***
```

Figura 6-4. Ejecución Enlace 6-3.

Tras finalizar la simulación comprobamos que, en esta, a diferencia de la anterior, el programa nos pide menos datos, esto ocurre porque, según la programación que presenta, si se cumple la condición $h > h_{req}$ directamente, las pérdidas serán 0.

Se verifica nuevamente que los datos que se obtienen como respuesta son correctos y similares a lo que se calcularon en apartados anteriores.

6.2 Medio con obstáculos

6.2.1 Caso 1: Un obstáculo “Filo de Cuchillo”

Cuando se trata de un obstáculo aislado, es complejo encontrar un perfil en el que se pueda ver claramente que es este el que obstruye sin cancelar la comunicación entre antenas. El perfil que se ha encontrado y utilizado para probar el programa nos deja ver un obstáculo que cumple las condiciones que se buscan. Las antenas se encuentran en:

TX:

Latitud: 40.789
Longitud: -3.03

RX:

Latitud: 40.786
Longitud: -3.03



Figura 6-5. Enlace y perfil elevación Google Earth.

En la simulación se comprueba que con los mismos datos que se utilizaron en el apartado 4.2.1, el enlace solo presenta un obstáculo.

```

Bienvenido al simulador de radioenlaces con obstáculos. Se necesitan los siguientes datos para el estudio:
Frecuencia en Hz: 2e9
Altura de la antena TX (mástil) en m: 7
Altura de la antena RX (mástil) en m: 7
Radio efectivo de la tierra en km: 6370
Introduce la latitud de la antena TX en formato decimal: 40.789
Introduce la longitud de la antena TX en formato decimal: -3.03
Introduce la latitud de la antena RX en formato decimal: 40.786
Introduce la longitud de la antena RX en formato decimal: -3.03
Este enlace presenta 1 obstáculo que produce pérdidas
¿Se conoce el radio del obstáculo? s o n: n
Las pérdidas producidas por el obstáculo que se encuentra en las coordenadas ( 40.7874308617 , -3.03 ) y cuya altura es de 831.325256348 m
toman un valor de: 6.66299726742 dB

***Repl Closed***
    
```

Figura 6-6. Ejecución Enlace 6-5.

Además de dar las pérdidas que se producen en este tipo de enlaces, el programa indica las coordenadas exactas donde se encuentra el pico del obstáculo que está produciéndolas, su altura respecto al nivel del mar y las pérdidas.

Al igual que en el resto de ejemplos, los decimales y la cierta variación de resultados se producen como consecuencia de los redondeos, que al dar resultados en *dBs* se ven incrementadas.

6.2.2 Caso 2: Un obstáculo con radio conocido

El perfil que se va a tomar para este caso es el mismo que el anterior. Recordando:

TX:

Latitud: 40.789

Longitud: -3.03

RX:

Latitud: 40.786

Longitud: -3.03



Figura 6-7. Enlace y perfil elevación Google Earth.

Se toma como radio del obstáculo $20m$, como se indicó en la resolución analítica, y una vez más se comprueba que el código es totalmente válido.

```

Bienvenido al simulador de radioenlaces con obstáculos. Se necesitan los siguientes datos para el estudio:
Frecuencia en Hz: 2e9
Altura de la antena TX (mástil) en m: 7
Altura de la antena RX (mástil) en m: 7
Radio efectivo de la tierra en km: 6370
Introduce la latitud de la antena TX en formato decimal: 40.789
Introduce la longitud de la antena TX en formato decimal: -3.03
Introduce la latitud de la antena RX en formato decimal: 40.786
Introduce la longitud de la antena RX en formato decimal: -3.03
Este enlace presenta 1 obstáculo que produce pérdidas
¿Se conoce el radio del obstáculo? s o n: s
Introduzca el valor del radio en metros: 20
Las pérdidas producidas por el obstáculo que se encuentra en las coordenadas ( 40.7874308617 , -3.03 ) y cuya altura es de 831.325256348 m
toman un valor de: 11.3762106208 dB

***Repl Closed***

```

Figura 6-8. Ejecución Enlace 6-7.

6.2.3 Caso 3: Dos obstáculos aislados “Filo de Cuchillo”

Cuando se busca un enlace con dos obstáculos aislados se tiene que tener en cuenta que, la distancia entre ambos no puede ser inferior a $250m$, ya que, si es así, se nos mostrará un mensaje por pantalla diciendo que el método que calcula las pérdidas en este tipo de enlaces no se encuentra implementado.

El ejemplo que se presenta, presenta numerosos picos con posibilidad de producir obstrucción, pero jugando con las alturas de las antenas y con la frecuencia de trabajo, se ha conseguido que solo afecten al radioenlace los dos picos más altos tal y como se ven en la imagen que se muestra a continuación.

Las coordenadas que se utilizarán son:

TX:

Latitud: 61.463363

Longitud: 6.962707

RX:

Latitud: 61.457852

Longitud: 7.603182

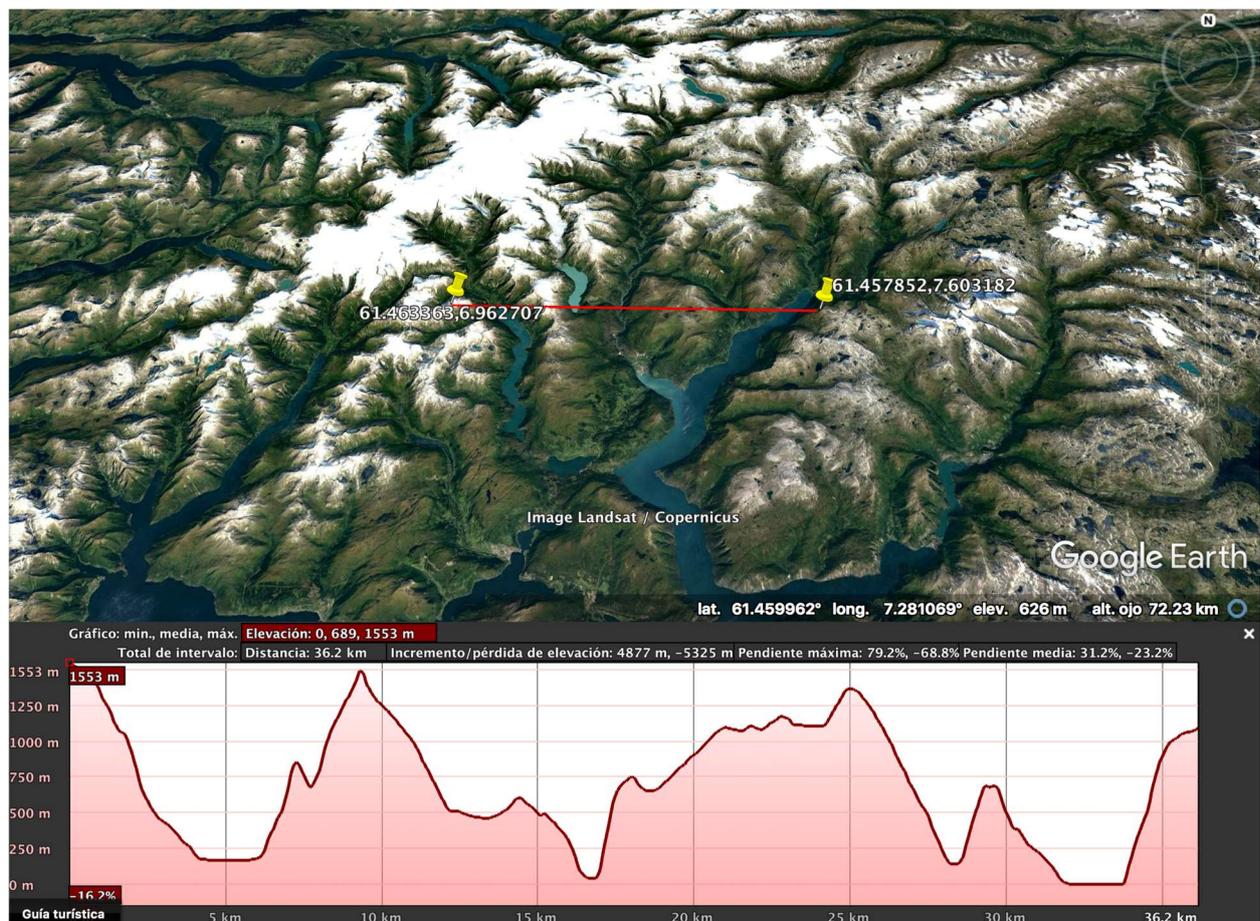


Figura 6-9. Enlace y perfil elevación Google Earth.

Tras realizar la prueba con el programa se obtiene lo que se muestra:

```

Bienvenido al simulador de radioenlaces con obstáculos. Se necesitan los siguientes datos para el estudio:
Frecuencia en Hz: 1e9
Altura de la antena TX (mástil) en m: 5
Altura de la antena RX (mástil) en m: 3
Radio efectivo de la tierra en km: 6370
Introduce la latitud de la antena TX en formato decimal: 61.463363
Introduce la longitud de la antena TX en formato decimal: 6.962707
Introduce la latitud de la antena RX en formato decimal: 61.457852
Introduce la longitud de la antena RX en formato decimal: 7.603182
Este enlace presenta 2 obstáculos que producen pérdidas
¿Se conoce el radio del obstáculo? s o n: n
Las pérdidas producidas por los obstáculos que se encuentran en las coordenadas ( 61.4622359087 , 7.12701782856 ) y ( 61.4599098953 , 7.40169507425 ) cuyas alturas son de 1496.35241699 m y 1364.5222168 m toman un valor de: 31.6197182977 dB

***Repl Closed***

```

Figura 6-10. Ejecución Enlace 6-9.

En este caso el resultado obtenido difiere unos 2dB aproximadamente del resultado que se tuvo en la resolución analítica, este error puede estar producido por la diferencia entre la distancia total del enlace (de 36km a 36.2km) del redondeo de la distancia al primer obstáculo (en el enlace de algo menos de 14.5km) y del resto de redondeos que se producen en las ecuaciones.

6.2.4 Caso 4: Dos obstáculos aislados con radios conocidos

El enlace es el mismo que el utilizado anteriormente, pero con radios conocidos igual a 15m y 25m como se propuso en el apartado 4.

```

Bienvenido al simulador de radioenlaces con obstáculos. Se necesitan los siguientes datos para el estudio:
Frecuencia en Hz: 1e9
Altura de la antena TX (mástil) en m: 5
Altura de la antena RX (mástil) en m: 3
Radio efectivo de la tierra en km: 6370
Introduce la latitud de la antena TX en formato decimal: 61.463363
Introduce la longitud de la antena TX en formato decimal: 6.962707
Introduce la latitud de la antena RX en formato decimal: 61.457852
Introduce la longitud de la antena RX en formato decimal: 7.603182
Este enlace presenta 2 obstáculos que producen pérdidas
¿Se conoce el radio del obstáculo? s o n: s
Introduzca el valor del radio del primer obstáculo en metros: 15
Introduzca el valor del radio del segundo obstáculo en metros: 25
Las pérdidas producidas por los obstáculos que se encuentran en las coordenadas ( 61.4622359087 , 7.12701782856 ) y ( 61.4599098953 , 7.40169507425 ) cuyas alturas son de 1496.35241699 m y 1364.5222168 m toman un valor de: 33.1607628994 dB

***Repl Closed***

```

Figura 6-11. Ejecución Enlace 6-9.

Al igual que el caso anterior, la diferencia entre el resultado obtenido con la ejecución del programa y la analítica difiere unos 3dB s consecuencia de nuevo del redondeo en los datos y en las soluciones.

6.2.5 Caso 5: Obstáculo que hace inviable el radioenlace

Para este caso, el enlace que se presenta en Google Earth ya hace pensar que va a ser inviable, tal y como se verifica después.

Las coordenadas utilizadas son:

TX:

Latitud: 61.759726

Longitud: 12.095607

RX:

Latitud: 61.749469

Longitud: 12.126524

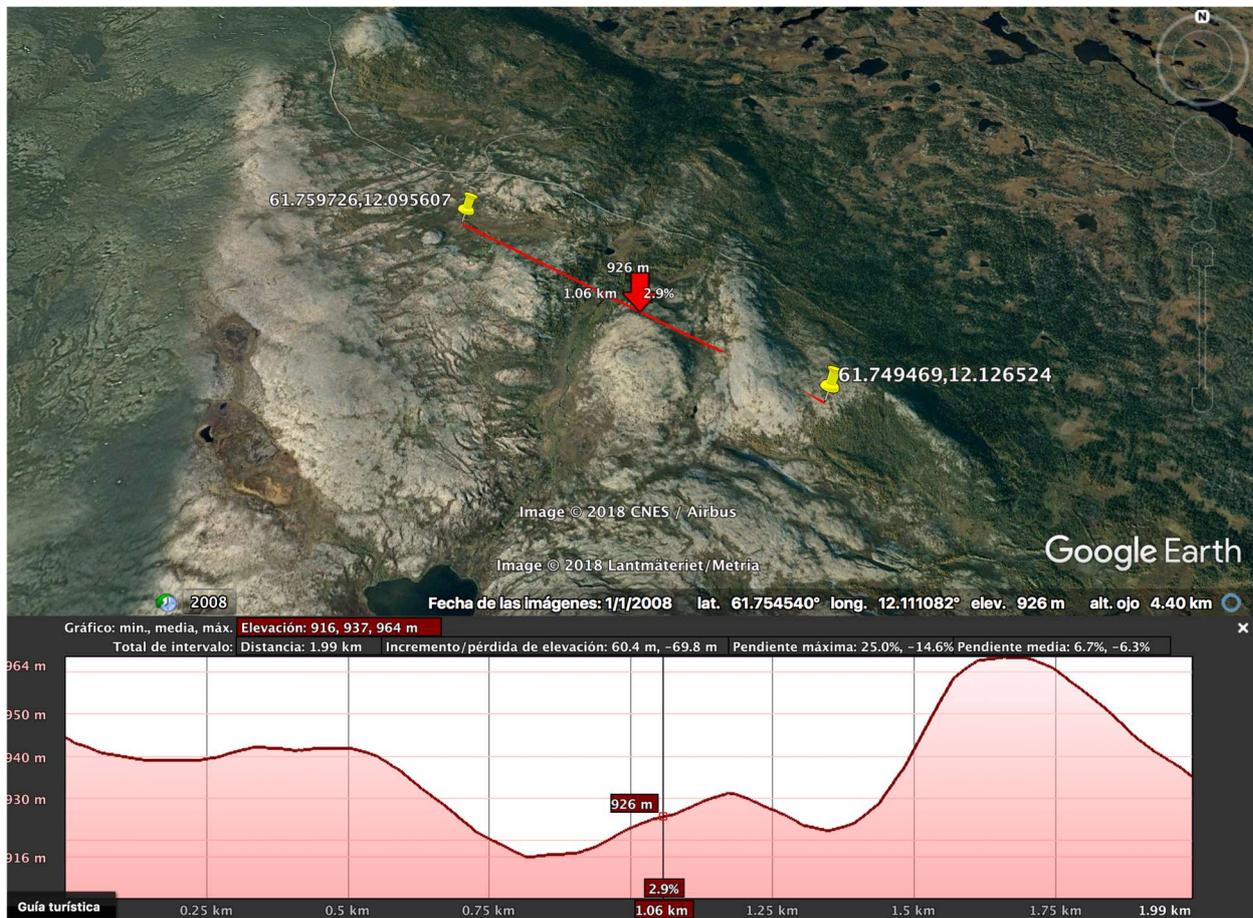


Figura 6-12. Enlace y perfil elevación Google Earth.

Procediendo al a ejecución del código, se obtiene lo esperado. Independientemente del número de obstáculos que pueda tener el enlace, en el momento que uno de ellos obstruye más del 60% de la primera zona de Fresnel en ese punto, el programa finaliza diciendo esto junto con las coordenadas del pico que no permite la comunicación. Si hubiera más de uno, presentará el primero de ellos.

```

Bienvenido al simulador de radioenlaces con obstáculos. Se necesitan los siguientes datos para el estudio:
Frecuencia en Hz: 2e9
Altura de la antena TX (mástil) en m: 10
Altura de la antena RX (mástil) en m: 10
Radio efectivo de la tierra en km: 6370
Introduce la latitud de la antena TX en formato decimal: 61.759726
Introduce la longitud de la antena TX en formato decimal: 12.095607
Introduce la latitud de la antena RX en formato decimal: 61.749469
Introduce la longitud de la antena RX en formato decimal: 12.126524
Este enlace no es viable. El primer elipsoide de Fresnel se encuentra obstruido mas del 60% en el punto ( 61.7511755563 , 12.1213829215 )

***Repl Closed***
    
```

Figura 6-13. Ejecución Enlace 6-12.

7 CONCLUSIONES

*Serio retrato en la pared clarea
todavía. Nosotros divagamos.
En la tristeza del hogar golpea
el tictac del reloj.
Todos callamos.*

- Antonio Machado -

Durante la realización de este trabajo he puesto en práctica numerosos conocimientos que he ido adquiriendo en el Grado en estos años, por tanto, me ha servido para afianzarlos, ampliarlos y terminar de comprender aquello que no quedó claro. Además de demostrarme que los puedo tratar con total soltura y confianza.

He afianzado mis conocimientos en Python y he comprobado que han mejorado de forma exponencial llegando a realizarse un código robusto y estructurado capaz de implementar lo que se pedía, es decir, los objetivos se han cumplido.

El principal problema que he encontrado ha sido cuando tuve que asumir que el terreno real no es, ni parecido, a lo ideal que se trata en la resolución matemática. En un principio, en un enlace de corto alcance, donde claramente se veía que solo podría obstruir un pico, me encontraba que el programa decía que había más de 70 u 80. No era mentira, los había, pero ¿dónde estaba el fallo? Lo que ocurría era que, en un principio se analizaban todas y cada una de las 500 muestras que se cogen entre una punta y otra del enlace, una por una se calculaban distancias, despejamientos y se comprobaba si obstruía o no la primera zona de Fresnel en ese punto. Por tanto, desde que llegaba a determinada altura (la ladera iba subiendo) hasta el pico máximo y luego hasta llegar otra vez de nuevo a otra altura (en la bajada) todos los puntos obstruían en mayor o menor medida y se contabilizaban como obstáculos. Esto fue resuelto con el bucle que se realiza en la función *num_obs* que solo guarda en la tabla de *posibles* aquellos puntos cuyo anterior y siguiente son inferiores, es decir, los picos máximos del obstáculo y solo es ese punto en sus coordenadas el que se analiza frente al radio de Fresnel.

Tras la resolución de este y del resto de pequeños problemas que iban surgiendo, se ha conseguido que este código funcione con precisión a la misma vez que rápido y sin consumir prácticamente ningún recurso de la máquina en la cual se está ejecutando.

Si se mira a futuro, el código que aquí se ha implementado, puede servir como base y parte para la realización de un programa capaz de calcular las pérdidas de N obstáculos con N muestras del terreno, implementando otros métodos como el de la geometría multipunto o el de Bullintong. Se puede seguir avanzando, también, si no solo se calculan las pérdidas por difracción, ya que, teniendo la geolocalización exacta del enlace se pueden añadir pérdidas por porcentaje de lluvia, nieve u otras adversidades meteorológicas.

La oportunidad de crear una interfaz gráfica para este u otro proyecto mayor es algo que quedó pendiente y que sería un acierto total, haciendo que su uso pueda verse incrementado al facilitarse al gran público.

Por tanto, el proyecto realizado como trabajo fin de grado ha supuesto para mí un inicio del que pueda ser mi futuro, uniendo la radiocomunicación con un lenguaje que realmente me ha aportado seguridad a la hora de programar.

Además, me ha dado la oportunidad de sentir la satisfacción de poder aportar mis conocimientos, mi trabajo y mi dedicación para futuro uso en la escuela dónde alguien, algún día, se encargó de enseñármelos.

REFERENCIAS

- [1] «Code Academy,» [En línea]. Available: <https://www.codecademy.com>.
- [2] Radiocommunication Sector of ITU, «Recommendation ITU-R P.526-14, Propagation by diffraction,» 2013.
- [3] «Electromagnetismo III,» 2011. [En línea]. Available: <http://ema315.blogspot.com.es/2010/09/zonas-de-fresnel-en-redes-inalambricas.html>.
- [4] Radiocommunication Sector of ITU, «Definition of penumbra width - FIGURE 1 (ITU-R P.526-14),» 2013.
- [5] Radiocommunication Sector of ITU, «REC. P.530-17. Propagation data and prediction methods required for the design of terrestrial line-of-sight systems,» 2017.
- [6] Radiocommunication Sector of ITU, «REC. P310-9*,» 1994.
- [7] Radiocommunication sector of ITU, «Path clearance - FIGURE 7 (ITU-R P.526-14),» 2013.
- [8] Radiocommunication sector of ITU, «Knife-edge diffraction loss - FIGURE 9 (ITU-R P.526-14),» 2013.
- [9] Radiocommunication sector of ITU, «Recommendation ITU-R P.526-14, Propagation by diffraction,» 2013.
- [10] Radiocommunication sector of ITU, «Method for double isolated edges - FIGURE 11 (ITU-R P.526-14)».
- [11] Radiocommunication sector of ITU, «Figure showing the main and the second obstacle - FIGURE 12 (ITU-R P.526-14),» 2013.
- [12] Radiocommunication Sector of ITU, «Rec. UIT-R P.527-3».
- [13] Python ORG, [En línea]. Available: <https://www.python.org/doc/>.
- [14] B. S. Marco, «Introducción a la programación con Python,» 2018. [En línea]. Available: <http://www.mclibre.org/consultar/python/lecciones/python-if-else.html>.
- [15] 2004. [En línea]. Available: <http://www.espaciolinux.com/foros/programacion/como-llamar-una-funcion-creada-por-usuario-python-desde-otro-script-t50317.html>.
- [16] Google, «Google Maps API,» [En línea]. Available: <https://developers.google.com/maps/documentation/distance-matrix/intro?hl=es-419#DistanceMatrixRequests>.
- [17] dh--, «GitHub, Inc.,» 2013. [En línea]. Available: <https://github.com/googlemaps/js-v2-samples/blob/gh-pages/elevation/ElevationParser.py>.
- [18] B. S. Marco, «Introducción a la programación con Python,» 2014. [En línea]. Available: <http://www.mclibre.org/consultar/python/lecciones/python-listas.html>.
- [19] B. S. Marco, «Introducción a la programación con Python,» [En línea]. Available: <http://www.mclibre.org/consultar/python/lecciones/python-for.html>.
- [20] «MathWorks,» [En línea]. Available: <https://es.mathworks.com/help/matlab/ref/integral.html>.
- [21] D. Rock, «Donnierock,» 2015. [En línea]. Available: <https://donnierock.com/2015/06/29/dar-formato-a-un-float-para-mostrar-solo-dos-decimales-en-python/>.

- [22] S. Ramírez, «Python Blog,» 2016. [En línea]. Available:
<http://pythoninicios.blogspot.com.es/2016/03/como-calcular-la-distancia-entre-dos.html>.
- [23] Google, «Google Earth,» [En línea]. Available:
<https://www.google.es/intl/es/earth/index.html>.

GLOSARIO

ITU-R: Normas técnicas internacionales desarrolladas por el Sector de Radiocomunicaciones de la UIT. Éstas son el resultado de estudios efectuados por las Comisiones de Estudio de Radiocomunicaciones sobre la utilización de una amplia gama de servicios inalámbricos; la gestión del espectro de radiofrecuencia y las órbitas de satélite el uso eficaz del espectro de radiofrecuencia por todos los servicios de radiocomunicaciones; la radiodifusión terrenal y las radiocomunicaciones por satélite; la propagación de las ondas radioeléctricas; los sistemas y las redes para el servicio fijo por satélite; y las operaciones espaciales, el servicio de exploración de la Tierra por satélite, el servicio de meteorología por satélite y el servicio de radioastronomía.

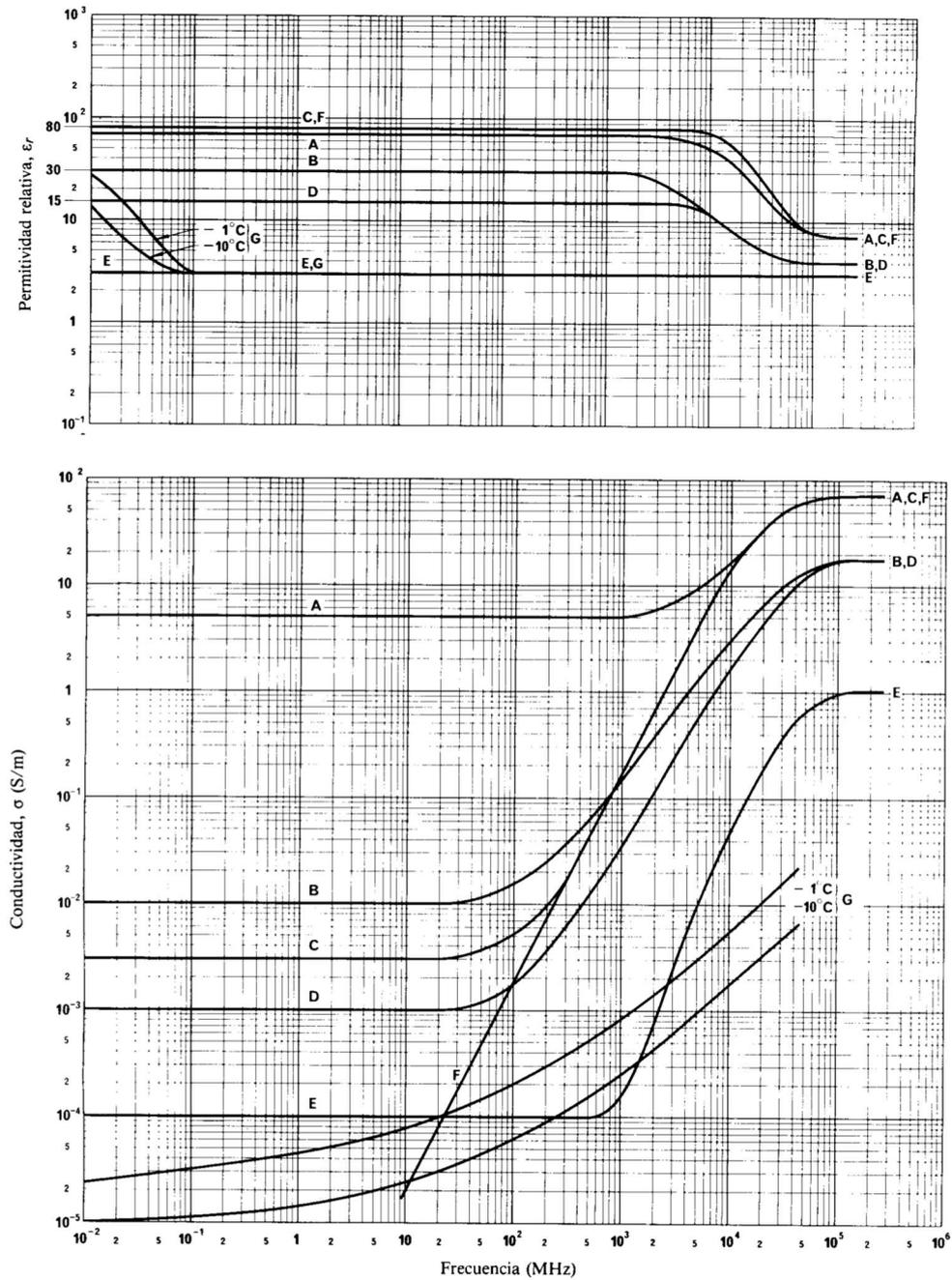
Difracción: Desviación de una onda al chocar con el borde de un cuerpo opaco o al atravesar una abertura.

LoS: Line of Sight. Característica de la radiación electromagnética o de la propagación de ondas acústicas, que significa que las ondas viajan en una ruta directa desde la fuente hasta el receptor.

Dispersión troposférica: Fenómeno donde las ondas de radio a frecuencias UHF y SHF se dispersan aleatoriamente a medida que atraviesan las capas superiores de la troposfera. A medida que las señales pasan a través de la troposfera, parte de la energía se dispersa hacia la Tierra, permitiendo que la estación receptora capte la señal.

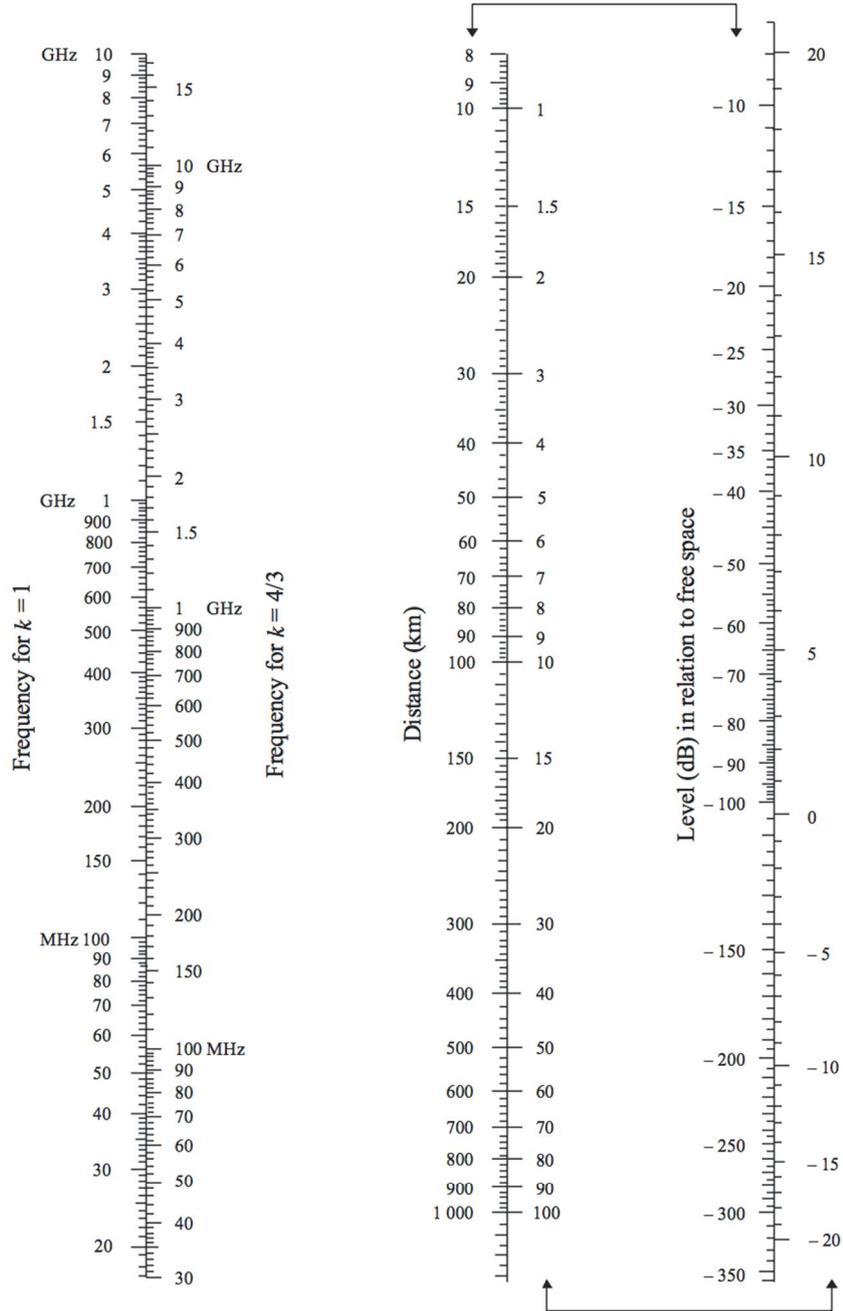
Debido a que la troposfera es turbulenta y tiene una alta proporción de humedad, las señales de radio de dispersión troposférica son refractadas y, en consecuencia, solo una pequeña proporción de la energía de radio es recolectada por las antenas receptoras. Las frecuencias de transmisión alrededor de 2 GHz son las más adecuadas para los sistemas de dispersión troposférica ya que a esta frecuencia mejoran las relaciones señal / ruido.

FIGURA 1
Permitividad relativa, ϵ_r , y conductividad, σ , en función de la frecuencia



- A: Agua salada (salinidad media), 20°C
- B: Suelo húmedo
- C: Agua dulce, 20°C
- D: Suelo moderadamente seco
- E: Suelo muy seco
- F: Agua pura, 20°C
- G: Hielo (agua dulce)

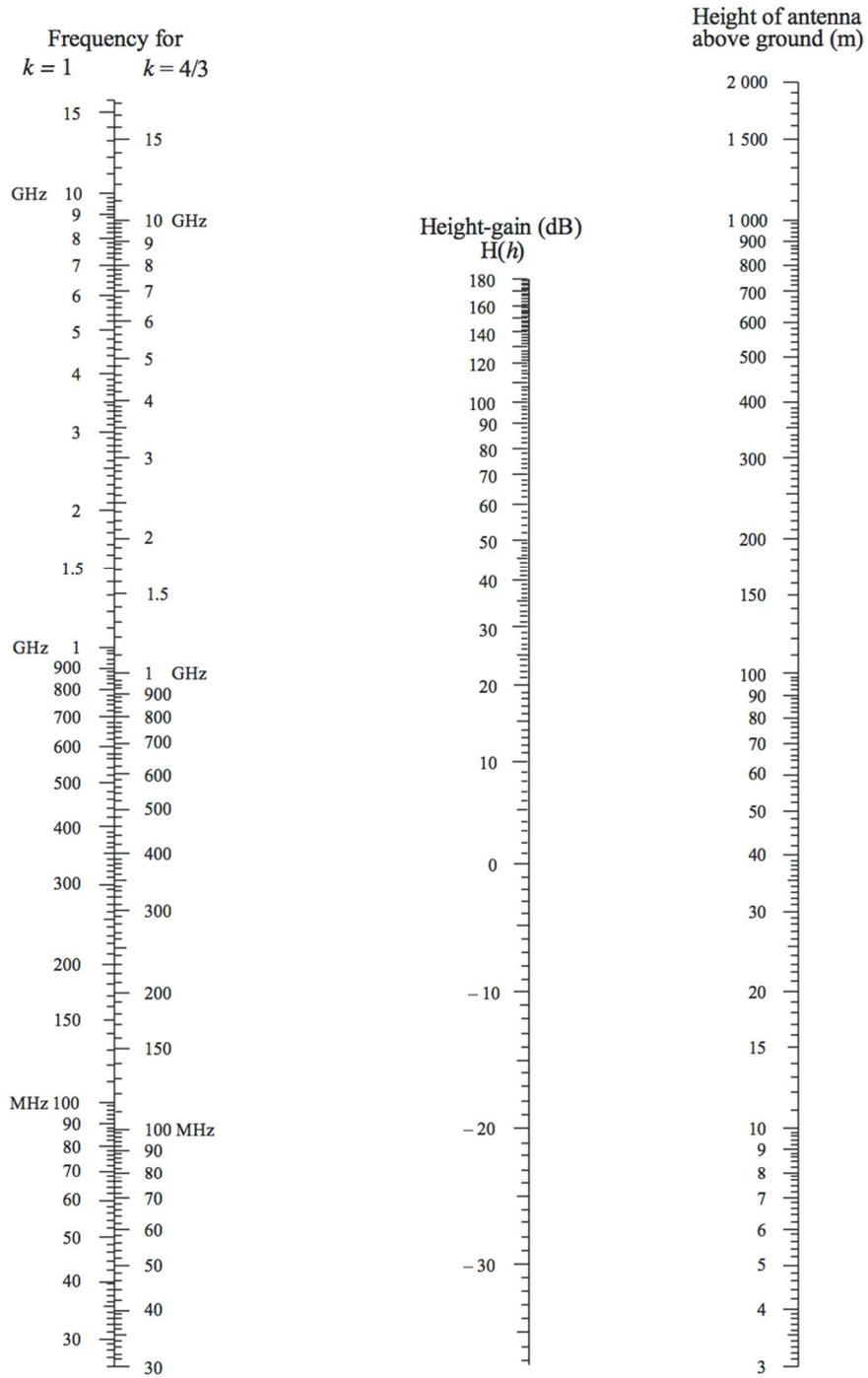
Diffraction by a spherical Earth – effect of distance



Horizontal polarization over land and sea
Vertical polarization over land

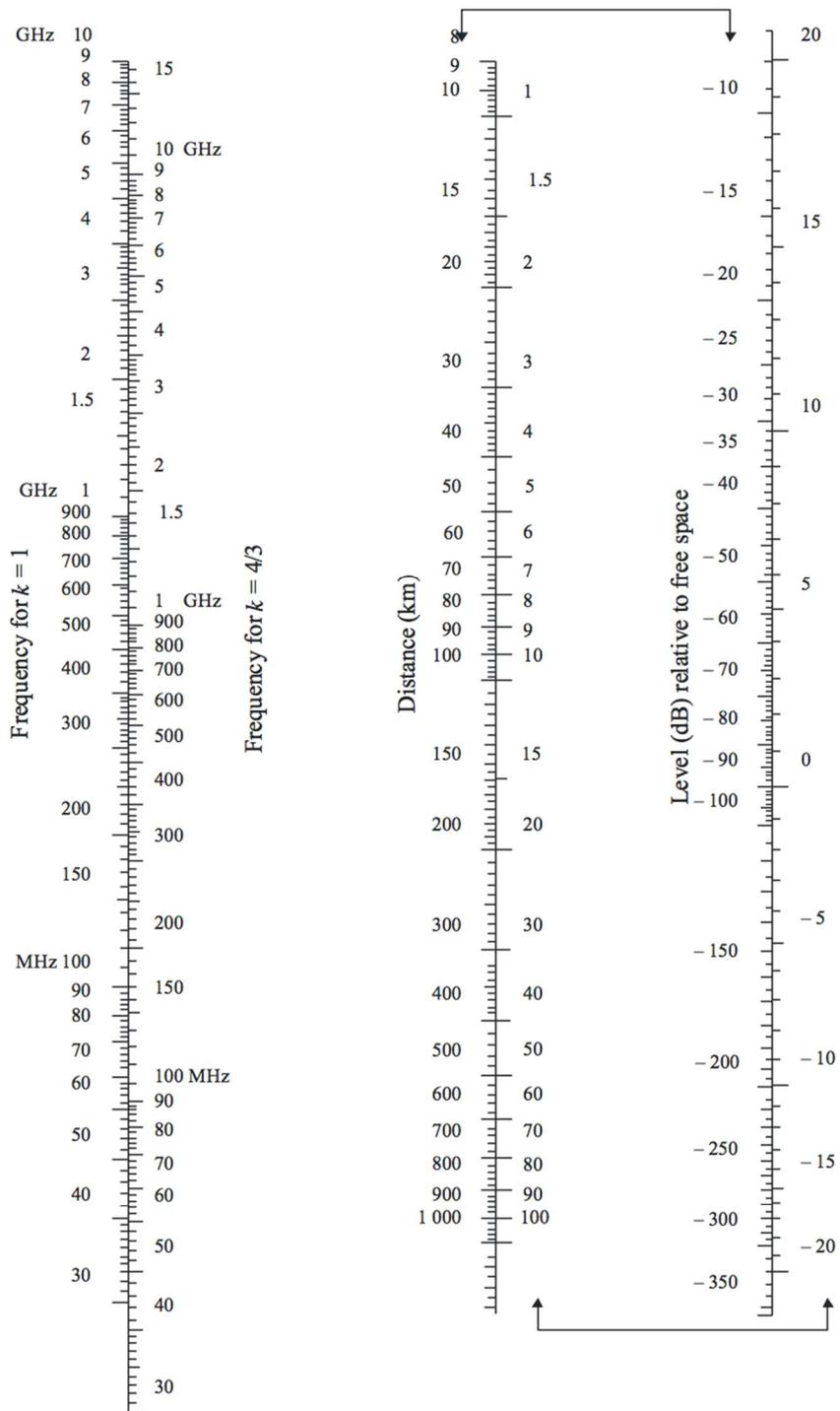
(The scales joined by arrows should be used together)

Diffraction by a spherical Earth – height-gain



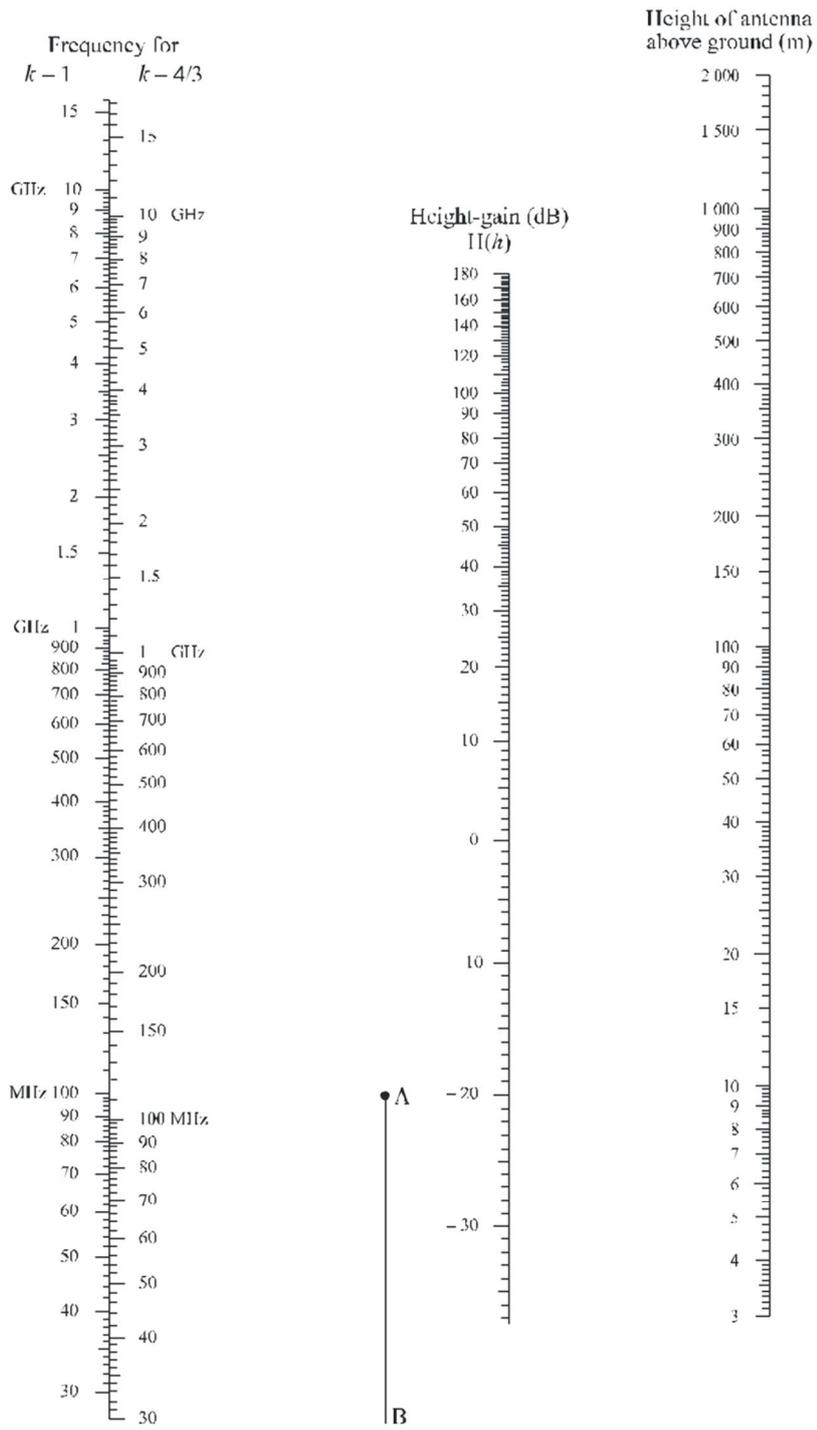
Horizontal polarization – land and sea
 Vertical polarization – land

Diffraction by a spherical Earth – effect of distance



Vertical polarization over sea
(The scales joined by arrows should be used together)

Diffraction by a spherical Earth – height-gain



Vertical polarization – sea

La Google Maps Elevation API ofrece la oportunidad de realizar consultas de ubicaciones en la superficie terrestre y obtener datos de elevación. Proporciona datos de elevación para todas las ubicaciones sobre la superficie terrestre, incluidas ubicaciones profundas en el lecho marino (que devuelven valores negativos). En aquellos casos en los cuales Google no cuente con mediciones de elevación exactas para la ubicación precisa de tu solicitud, el servicio realizará una interpolación y devolverá un valor promediado recurriendo a las cuatro ubicaciones más cercanas. Los valores de elevación se expresan en función del nivel medio del mar (NMM) local.

Se puede acceder a la API a través de una interfaz HTTP, como se ha realizado en este proyecto.

Solicitudes de elevación

Las solicitudes de Google Maps Elevation API se construyen como una string de URL. Se deben especificar los datos de ubicación en una de estas dos maneras:

- Como un conjunto de una o más `locations`.
- Como una serie de puntos conectados a lo largo de una `path`.

Cualquiera de estos enfoques usa coordenadas de latitud/longitud para identificar las ubicaciones o los vértices del trayecto.

Una solicitud de Google Maps Elevation API debe respetar la siguiente forma que se encuentra entre los corchetes, pues las URLs deben estar correctamente codificadas para ser válidas:

- `outputFormat` puede ser cualquiera de los siguientes valores: `json` o `xml`

Si no es posible usar HTTPS, usa lo siguiente para acceder a la Google Maps Elevation API a través de HTTP:

Parámetros de la solicitud

Las solicitudes a la Google Maps Elevation API utilizan diferentes parámetros en función de si la solicitud es para ubicaciones discontinuas o para un trayecto ordenado.

Como en todas las direcciones URL, los parámetros se separan con el carácter `&`.

- `key` (*obligatorio*) es la clave de API de tu aplicación. Esta clave identifica tu aplicación a los fines de la administración de la cuota. Cómo conseguir una clave se explica posteriormente.
- `locations` (*obligatorio*) define las ubicaciones terrestres a partir de las cuales deben devolverse datos de elevación. Este parámetro toma una ubicación individual como par de `{latitude,longitude}` separado por comas (`locations=40.714728,-73.998672`) o múltiples pares de latitud/longitud como una matriz (`locations=40.714728,-73.998672|-34.397,150.644`) o una polilínea codificada (`locations=enc:gfo}EtohhU`). Las latitudes pueden tener cualquier valor que para ser válido tiene que estar entre `-90` y `90` mientras que los valores de longitud pueden

tener cualquier valor entre `-180` y `180`. Si se especifica un valor de latitud o longitud no válido, se rechazará la solicitud por ser incorrecta.

- `path` (*obligatorio*) define un trayecto en el planeta para el cual deben devolverse datos de elevación. Este parámetro define un conjunto de dos o más pares ordenados de `{latitud,longitud}`(`path=40.714728,-73.998672|-34.397,150.644`). Este parámetro se debe usar junto con el parámetro `samples`.
- `samples` (*obligatorio*) especifica la cantidad de puntos de ejemplo a lo largo de un trayecto para el cual deben devolverse datos de elevación. Divide el parámetro `path` en un conjunto de puntos equidistantes a lo largo del trayecto.

Respuestas de elevación

Cada respuesta contendrá los siguientes elementos:

- `status` de elevación, que puede ser uno de los siguientes:
 - `OK` indica que la solicitud de API se realizó con éxito.
 - `INVALID_REQUEST` indica que el formato de la solicitud era incorrecto.
 - `OVER_QUERY_LIMIT` indica que el solicitante excedió su cuota.
 - `REQUEST_DENIED` indica que la API no completó la solicitud.
 - `UNKNOWN_ERROR` indica error desconocido.
- `Results`, matriz que incluye:
 - `location` (contiene `lat` y `lng`) de la posición para la cual se están calculando los datos de elevación. Para las solicitudes de trayectos, el conjunto de elementos `location` contendrá los puntos muestreados a lo largo de ese trayecto.
 - `elevation` que indica la elevación de la ubicación en metros.
 - `resolution` indica la distancia máxima entre puntos de datos a partir de los cuales se interpoló la elevación, en metros. Esta propiedad estará ausente si se desconoce la resolución.

Obtener una clave de autenticación

Todas las aplicaciones de Google Maps Elevation API requieren autenticación.

Si se usa la API con el plan estándar, se debe emplear una clave de API configurada en el proyecto que se elija. Incluir una clave en la solicitud permite controlar el uso de la API por parte de la aplicación y habilita el acceso a una cuota diaria gratuita.

Para obtener una clave válida hay que entrar en la Google API Console con nuestro correo electrónico de Gmail y crear o seleccionar un proyecto. Tras esto, hay que habilitar la API en el proyecto en el cual nos encontramos, dentro de este, en la página “Credentials” se obtiene la clave y se configuran sus restricciones.

También se puede buscar una clave existente en Google API Console.

La clave que se ha usado en este trabajo esta asignada al proyecto TFG del email:

correo_personal@gmail.com.

Límites que se aplican a Google Maps Elevation API con uso estándar

- 2500 solicitudes gratuitas por día, calculadas como la suma de las consultas del cliente y el servidor.
- 512 ubicaciones por solicitud.
- 50 solicitudes por segundo, calculadas como la suma de las consultas del cliente y el servidor.

ANEXO IV

Este código está disponible para su descarga en la siguiente URL:

<https://github.com/merydiou/Simulador-Radioenlaces-Python-TFG>

TFG_Principal.py

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import sys
from Terreno import *
from TerrenoLlano import *
from TerrenoObstaculos import *

#TODO EL CÓDIGO Y FUNCIONES EXTERNAS ESTAN PROGRAMADAS PARA PYTHON 2.7

c=float(3e8)

print "Bienvenido al simulador de radioenlaces con obstáculos. Se necesitan los
siguientes datos para el estudio:"
#frecuencia
freq = float(raw_input("Frecuencia en Hz: "))
h_TXm = float(raw_input("Altura de la antena TX (mástil) en m: "))
h_RXm = float(raw_input("Altura de la antena RX (mástil) en m: "))
#radio efectivo de la tierra
a_e= float(raw_input("Radio efectivo de la tierra en km: "))

latTX = raw_input('Introduce la latitud de la antena TX en formato decimal: ')
lonTX = raw_input('Introduce la longitud de la antena TX en formato decimal: ')
latRX = raw_input('Introduce la latitud de la antena RX en formato decimal: ')
lonRX = raw_input('Introduce la longitud de la antena RX en formato decimal: ')

#wave lenght
wave_len=c/freq

#numero de muestras que se cogen en el enlace
muestras=500
#Se llama a la funcion que nos devuelve el perfil del enlace y del cual se obtendra la
elevacion en cada punto asi como su latitud y longitud.
hobst_lat_lon=perfil_terreno(latTX,lonTX,latRX,lonRX,muestras)

elevacion=hobst_lat_lon[0]
latitud=hobst_lat_lon[1]
longitud=hobst_lat_lon[2]

"""Esta funcion imprime un enlace que lleva a una grafica del perfil que se esta
estudiando, esta comentada porque no se hace uso de ella,
pero se deja propuesta por si hiciera falta en un futuro:
#print getChart(chartData=elevacion)
"""

#Distancia total del enlace
dis=GetDistance(latitud[0],longitud[0],latitud[muestras-1],longitud[muestras-1],a_e)
#Si la antena trasmisora y receptora estan en el mismo punto
if dis==0:
    print "ERROR: La distancia entre ambas antenas es 0."
    #Salimos del programa y se finaliza
    sys.exit()
#Se comprueba cual de las dos antenas se encuentra mas alta
#En el caso de ser la primera
if elevacion[0]>elevacion[muestras-1]:
```

```

        #Se modifican las alturas tomando como 0 absoluto la antena que se encuentra a
menor altura
        h_RX=h_RXm
        h_TX=h_TXm+elevacion[0]-elevacion[muestras-1]
        elevacion_abs=elevacion[muestras-1]
#En otro caso
else:
        #Se modifican las alturas tomando como 0 absoluto la antena que se encuentra a
menor altura
        h_TX=h_TXm
        h_RX=h_RXm+elevacion[muestras-1]-elevacion[0]
        elevacion_abs=elevacion[0]

#Se modifican el resto de elevaciones del terreno para que todas esten bajo el mismo
sistema de referencia
for n in range(len(elevacion)):
        elevacion[n]=elevacion[n]-elevacion_abs

#La altura final de las antenas es la calculada anteriormente, hay que sobrescribir
en sus posiciones correspondientes
elevacion[0]=h_TX
elevacion[muestras-1]=h_RX
        # Create the chart passing the array of elevation data

#tipo de terreno
tierra=terrain_type(dis,elevacion,wave_len,a_e,h_TX,h_RX,latitud,longitud,muestras)

#Si el medio que nos encontramos es llano y su frecuencia es mayor a 10MHz
if tierra=="medio_llano" and freq>=10e6 and a_e>0:
        perdidas_total=medio_llano(a_e,h_TX,h_RX,dis,wave_len,freq)
        #Se imprime el resultado
        print "Las pérdidas totales producidas por la difracción en este radioenlace
que no se encuentra obstruido por ningun obstáculo, son en dB:",perdidas_total
#Si el medio que nos encontramos es llano y su frecuencia es menor a 10MHz se
utilizará el programa de la ITU-R GRWAVE
elif (freq<=10e6 or a_e<=0) and tierra=="medio_llano":
        print("Utiliza el programa GRWAVE de la ITU-R ya que la frecuencia de trabajo
es menor a 10MHz y el medio a tratar es llano")
        #Se finaliza el programa
        sys.exit()

#Si el radioenlace se encuentra en un medio con obstaculos que obstreuyen se calculará
el numero de obstáculos y se procede
elif tierra=="medio_obstaculos":
        #Se calcula el numero de obstáculos
        obstaculos=numero_obstaculos(dis,latitud,longitud,elevacion,wave_len,a_e,muestr
as)

        """Esta función devuelve 6 parametros imprescindibles para el cálculo de las
pérdidas: el numero de obstaculos, la distancia de
estos al TX y al RX, la altura que tienen, su latitud y longitud, asi como el
despejamiento frente a la linea recta que une ambas
antenas"""
        num_obs=obstaculos[0]
        d1=obstaculos[1]
        d2=obstaculos[2]
        elev=obstaculos[3]
        lat=obstaculos[4]
        lon=obstaculos[5]
        despejamiento=obstaculos[6]

        if num_obs==0:
                perdidas_total=medio_llano(a_e,h_TX,h_RX,dis,wave_len,freq)

                #Se imprime el resultado
                print "Las pérdidas totales producidas por la difracción en este
radioenlace que no se encuentra obstruido por ningun obstáculo, son en
dB:",perdidas_total
        #Si el numero de obstaculos es 1
        elif num_obs==1:
                #Se calculan las perdidas que produce este obstaculo

```

```
    print "Este enlace presenta 1 obstáculo que produce pérdidas"
    perdidas=unico_obs(wave_len,dis,d1[0],d2[0],despejamiento[0])
    print "Las pérdidas producidas por el obstáculo que se encuentra en las
coordenadas (" ,lat[0]," ,",lon[0]," ) y cuya altura es de",elev[0]+elevacion_abs,"m
toman un valor de:",perdidas,"dB"
    #Si el numero de obstáculos es 2
    elif num_obs==2:
        print "Este enlace presenta 2 obstáculos que producen pérdidas"
        if GetDistance(lat[0],lon[0],lat[1],lon[1],a_e)<250:
            print "Los obstáculos que se encuentran en las coordenadas
(" ,lat[0]," ,",lon[0]," ) y (" ,lat[1]," ,",lon[1]," ) cuyas alturas son
de",elev[0]+elevacion_abs,"m y",elev[1]+elevacion_abs,"m están separados menos de 250m
por lo que no se puede aplicar el método de dos obstáculos aislados."
        else:
            #Se calculan las pérdidas que producen estos obstaculos

    perdidas=dos_obs(wave_len,dis,d1,d2,despejamiento,elev,lat,lon,a_e,h_RX)
    print "Las pérdidas producidas por los obstáculos que se
encuentran en las coordenadas (" ,lat[0]," ,",lon[0]," ) y (" ,lat[1]," ,",lon[1]," ) cuyas
alturas son de",elev[0]+elevacion_abs,"m y",elev[1]+elevacion_abs,"m toman un valor
de:",perdidas,"dB"
    #ERROR y se sale del programa
    elif num_obs=="error":
        sys.exit()
    #En el caso de ser mas de 2 obstáculos no se realiza nada
    else:
        print "El número de obstáculos de este enlace para esta frecuencia y
altura de antenas es de:",num_obs

if perdidas=="error":
    print "No ha introducido un valor correcto"
    sys.exit()
```


Terreno.py

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
from apiTFG import *
import math

#Funcion que devuelve la elevacion del terreno y la latitud y longitud de esos puntos
donde se mide
def perfil_terreno(latTX,lonTX,latRX,lonRX,muestras):
    #Se guardan en un formato concreto la latitud y la longitud para poder llamar a
    la API
    locTX=str(latTX)+", "+str(lonTX)
    locRX=str(latRX)+", "+str(lonRX)
    #Se unen las variables anteriores
    enlace = locTX + "|" + locRX
    #Tabla donde se va a guardar los resultados
    hobst_lat_lon=[]
    #Llamada a la API que nos devuelve los resultados del radioenlace
    hobst_lat_lon=getElevation(enlace,muestras)
    return hobst_lat_lon

#En esta funcion se decide que tipo de terreno es: llano o con obstaculos
def terrain_type(dis,elevacion, wave_len,a_e,h_TX,h_RX,latitud,longitud,muestras):
    """En este bucle se va analizando cada uno de los puntos en los que se ha
    medido el radioenlace y se analiza la distancia al TX,
    la distancia al RX y otros parametros para saber si obstruye o no"""
    for n in range(1,len(elevacion)-1,1):
        dis_TX=GetDistance(latitud[0],longitud[0],latitud[n],longitud[n],a_e)
        dis_RX=GetDistance(latitud[n],longitud[n],latitud[muestras-
1],longitud[muestras-1],a_e)
        #Se calcula el punto de la recta que une ambas antenas a esa distancia
        y_i=((h_RX-h_TX)/dis)*(dis_TX)+h_TX
        #Se calcula el despejamiento con la altura del punto y la elevacion del
        terreno
        despejamiento=elevacion[n]-y_i
        #Se calcula el radio del primer elipsoide de fresnel a esa distancia
        Fresnel_rad= math.sqrt((wave_len*dis_TX*dis_RX)/dis)
        #Si el despejamiento se encuentra entre -0.6 y 0.5 tendremos al menos un
        obstaculo
        if (despejamiento/Fresnel_rad>=-0.6) and
        (despejamiento/Fresnel_rad<=0.5):
            medio= "medio_obstaculos"
            break
        #Si obstruye mas del 0.5 habrá al menos un punto que no permite que el
        radioenlace sea viable
        elif despejamiento/Fresnel_rad>0.5:
            medio="medio_obstaculos"
            break
        #Si no obstruye, estaremos ante un medio llano
        else:
            medio="medio_llano"
    return medio
```


TerrenoLlano.py

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import math
import sys

def medio_llano(a_e,h_TX,h_RX,dis,wave_len,freq):
#Se calcula la distancia limite
    d_los=(math.sqrt(2*a_e*1e3))*(math.sqrt(h_TX)+math.sqrt(h_RX))
    #Si la distancia del enlace es mayor a la d_los se calcula K y la fuerza de
difraccion del campo en dBs
    if dis>=d_los:
        #Se calcula E en dBs
        K=admitancia_superficie(wave_len,a_e)

difraccion_campo_E=fuerza_difraccion_campo(K,dis,wave_len,a_e,freq,h_TX,h_RX)
        if K=="Error" or difraccion_campo_E=="Error":
            sys.exit()
        else:
            perdidas_total=difraccion_campo_E
    else:
        #Cuando d es menor que dlos se continua con el otro metodo y se imprimen
las perdidas
        perdidas_total=h_curvatura_rayo(dis,h_TX,h_RX,a_e,wave_len,freq)
    return perdidas_total

#Funcion que calcula K
def admitancia_superficie(wave_len, a_e):
    #polarizacion
    polarizacion=raw_input("Polarización de la antena TX(introducir h, para indicar
horizontal, o v, para vertical): ")
    #solo se tiene en cuenta el primer termino de la serie
    #Si la polarizacion es horizontal
    if polarizacion=="h" or polarizacion=="H":
        #Se tiene que introducir la permitividad y conductividad adecuadas a esa
frecuencia y medio
        permitividad_relativa=float(raw_input("Permitividad relativa: "))
        conductividad=float(raw_input("Conductividad en S/m: "))
        #Se calcula K
        K=((2*math.pi*a_e*1000)/wave_len)**(float(-
1)/float(3))*(((permitividad_relativa-1)**2+(60*wave_len*conductividad)**2)**(float(-
1)/float(4)))

        #Si la polarizacion es vertical
    elif polarizacion=="v" or polarizacion=="V":
        #Se tiene que introducir la permitividad y conductividad adecuadas a esa
frecuencia y medio
        permitividad_relativa=float(raw_input("Permitividad relativa: "))
        conductividad=float(raw_input("Conductividad en S/m: "))
        #Se calcula K_h y tras esto K
        K_h=((2*math.pi*a_e*1e3)/wave_len)**(float(-
1)/float(3))*(((permitividad_relativa-1)**2+(60*wave_len*conductividad)**2)**(float(-
1)/float(4)))
        K=K_h*math.sqrt(permitividad_relativa**2+(60*wave_len*conductividad)**2)

    #Si no se ha introducido un valor correcto
    else:
        print ("Introduce un valor correcto de polarización")
        K="Error"
    return K

#Funcion que calcula la fuerza de difraccion del campo
def fuerza_difraccion_campo(K, dis, wave_len, a_e, freq,h_TX,h_RX):
    #Si K fue error la fuerza de difraccion del campo tambien lo sera
    if K=="Error":
        f_dif_campo="Error"
    #Si K es menor que determinado valor, la fuerza sera 0
```

```

elif K<0.001:
    f_dif_camp=0
    #Si K es mayor que 1 habrá que utilizar el programa de la ITU GRWAVE
elif K>1:
    print("Utiliza el programa GRWAVE de la ITU-R ya que la frecuencia de
trabajo es menor a 10MHz y el medio a tratar es llano")
    f_dif_camp="Error"
    #En otro caso, se calculan el resto de parametros necesarios para saber la
fuerza de difraccion del campo
else:
    tierra_o_mar=raw_input("¿La distancia que separa las antenas es por
tierra o mar? Introduce t o w: ")
    #Si se introdujo un valor correcto
    if tierra_o_mar=="t" or tierra_o_mar=="T" or tierra_o_mar=="w" or
tierra_o_mar=="W":
        #Se llama a la funcion que calcula Beta, parametro imprescindible
para continuar
        beta=calc_beta(K,tierra_o_mar,freq)
        print "El parametro beta que depende del tipo de terreno toma el
valor:",beta
        #longitud normalizada del terreno entre ambas antenas
X=beta*dis*(math.pi/(wave_len*(a_e*1e3)**2))**(float(1)/float(3))
        #altura normalizada

Y_TX=2*beta*h_TX*((math.pi)**2/(wave_len**2*a_e*1e3))**(float(1)/float(3))
Y_RX=2*beta*h_RX*((math.pi)**2/(wave_len**2*a_e*1e3))**(float(1)/float(3))
        #termino que depende de la distancia
        if X<1.6:
            F=-20*math.log10(X)-5.6488*(X**float(1.425))
            print "F(X) en dB:",F
        else:
            F=11+10*math.log10(X)-17.6*X
            print "F(X) en dB:",F
        #Ganancia de alturas
G_TX=calc_G(Y_TX,beta,K)
        print "La ganancia de altura de la antena transmisora,G(Y_1), toma
el valor en dB:",G_TX
G_RX=calc_G(Y_RX,beta,K)
        print "La ganancia de altura de la antena receptora,G(Y_2), toma el
valor en dB:",G_RX
        f_dif_camp=F+G_TX+G_RX
        #Si el valor introducido no fue correcto
    else:
        f_dif_camp="Error"

return f_dif_camp

#Funcion que calcula el parámetro beta
def calc_beta(K, tierra_o_mar, freq):
    #Si el radioenlace es por tierra con una frecuencia mayor a 20MHz o por mar con
freq mayor a 300MHz, BETA=1
    if (((tierra_o_mar=="t" or tierra_o_mar=="T") and freq>=20e6) or
((tierra_o_mar=="w" or tierra_o_mar=="W" )and freq>=300e6)):
        beta=1
        #en otro caso se calcula
    else:
        beta=(1+1.6*K**2+0.67*K**4)/(1+4.5*K**2+1.53*K**4)
    return beta

#Funcion que calcula la ganancia de altura
def calc_G(Y,beta,K):
    #Parametro intermedio B que depende de beta y de Y_TX o Y_RX
B=beta*Y
    #Se aplica la formula de la ganancia de altura en funcion del valor de B
    if B<=2:
        ganancia_altura=20*math.log10(B+0.1*B**3)

```

```

else:
    ganancia_altura=17.6*(B-1.1)**(float(1)/float(2))-5*math.log10(B-1.1)-8

#Excepcion
if ganancia_altura<(2+20*math.log10(K)):
    ganancia_altura=2+20*math.log10(K)

return ganancia_altura

#Funcion que calcula los parametros necesarios para poder llamar a perdidas y devolver
ese valor
def h_curvatura_rayo(dis,h_TX,h_RX,a_e, wave_len,freq):
    #Parametros c y m necesarios para calcular b
    #c
    c_d=(h_TX-h_RX)/(h_TX+h_RX)
    #m
    m=(dis**2)/(4*a_e*1e3*(h_TX+h_RX))
    #b, necesario para calcular d_TX

    b=(2*math.sqrt(float(m+1)/float(3*m)))*math.cos((math.pi/3)+(float(1)/float(3))
    *math.acos((3*c_d/2)*math.sqrt((3*m)/((m+1)**3))))
    #distancia desde el transmisor hasta el punto donde la altura desde la
    curvatura de la tierra hasta el rayo es mas pequeña
    d_TX=(dis/2)*(1+b)
    #distancia desde el punto donde la altura desde la curvatura de la tierra hasta
    el rayo es mas pequeña hasta el RX
    d_RX=dis-d_TX
    #altura minima desde la curvatura de la tierra hasta el rayo que viaja entre
    ambas antenas
    h=((h_TX-(d_TX**2/(2*a_e*1e3)))*d_RX+(h_RX-(d_RX**2/(2*a_e*1e3)))*d_TX)/dis
    print b
    #con estos parametros se llama a la funcion perdidas
    return perdidas(h,d_TX,d_RX,wave_len,dis,h_TX,h_RX,freq)

#Funcion que calcula las perdidas que hay en el terreno
def perdidas(h,d_TX,d_RX,wave_len,dis,h_TX,h_RX,freq):
    #Se calcula h_req para poder compararlo con el h calculado con anterioridad
    h_req=0.552*math.sqrt((d_TX*d_RX*wave_len)/dis)
    #Si el valor de h introducido es mayor, no existen perdidas
    if h>h_req:
        perdidas=0
    #en otro caso, se calcula un a_e nuevo, modificado, y con esto, se calcula K y
    la fuerza de difraccion
    else:
        a_emodificado=0.5*(dis/(math.sqrt(h_TX)+math.sqrt(h_RX)))**2
        #se pasa a km para que en las formulas anteriores no de errores de
    unidades
        a_emodificado=a_emodificado/1e3
        print "El radio efectivo de la tierra modificado es de: ",a_emodificado,"
    km"
        K=admitancia_superficie(wave_len,a_emodificado)
        A_h=fuerza_difraccion_campo(K,dis,wave_len,a_emodificado,freq,h_TX,h_RX)
        #Si es menor que cero, no existen perdidas
        if A_h<=0:
            perdidas=0
        else:
            perdidas=A_h*(1-(h/h_req))
    return perdidas

```


TerrenoObstaculos.py

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import math
from apiTFG import *
from Cv_Sv import *

"""Funcion que calcula el numero de obstaculos y guarda la distancia al TX y al RX a la que se encuentra, la altura que tienen, su latitud y longitud, asi como el despejamiento frente a la línea recta que une ambas antenas"""
def numero_obstaculos(dis,latitud,longitud,elevacion,wave_len,a_e,muestras):
    #Variables
    n_obs=0
    latTX=latitud[0]
    lonTX=longitud[0]
    latRX=latitud[muestras-1]
    lonRX=longitud[muestras-1]
    h_TX=elevacion[0]
    h_RX=elevacion[muestras-1]
    #Tablas donde se guardarán los resultados que son devueltos
    d1=[]
    d2=[]
    elev=[]
    lat=[]
    lon=[]
    despej=[]
    #Listas donde se guardan los datos de los posibles obstáculos
    posibles=[[[],[],[]]

    """Bucle que recorre toda la tabla de elevacion y guarda en posibles aquellos puntos que son mayor que el punto anterior y mayor que el punto siguiente, es decir es un pico. Se guarda su altura, latitud y longitud"""
    for n in range(1,muestras-1,1):
        if elevacion[n]>elevacion[n-1] and elevacion[n]>elevacion[n+1]:
            posibles[0].append(elevacion[n])
            posibles[1].append(latitud[n])
            posibles[2].append(longitud[n])

    """Una vez filtrada la lista, con los valores de posibles se analiza cual de ellos obstruyen el radioenlace y solo aquellos que estan entre -0.6 y 0.5 e contarán como obstáculos. Si supera el 60% de obstrucción se devolverá error y se saldrá del programa, pues el radioenlace es inviable"""
    for b in range(len(posibles[0])):
        dis_TX=GetDistance(latitud[0],longitud[0],posibles[1][b],posibles[2][b],a_e)
        dis_RX=GetDistance(posibles[1][b],posibles[2][b],latitud[muestras-1],longitud[muestras-1],a_e)

        y_i=((h_RX-h_TX)/dis)*(dis_TX)+h_TX

        despejamiento=posibles[0][b]-y_i

        #Calculamos el radio del elipsoide de fresnel en ese punto
        Fresnel_rad= math.sqrt((wave_len*dis_TX*dis_RX)/dis)

        if despejamiento/Fresnel_rad>0.5:
            n_obs="error"
            print "Este enlace no es viable. El primer elipsoide de Fresnel se encuentra obstruido mas del 60% en el punto (" , posibles[1][b],",",posibles[2][b],")"
            break

        elif (despejamiento/Fresnel_rad>=-0.6) and (despejamiento/Fresnel_rad<=0.5):
```

```

        #Calculamos las distancias desde el TX al obstaculo y al RX desde el
obstaculo
        distancia1=GetDistance(latTX,lonTX,posibles[1][b],posibles[2][b],a_e)
        distancia2=GetDistance(posibles[1][b],posibles[2][b],latRX,lonRX,a_e)
        #Si la obstruye el numero de obstaculos se incrementará en 1, y se
guardaran las distancias calculadas
        n_obs+=1
        d1.append(distancia1)
        d2.append(distancia2)
        elev.append(posibles[0][b])
        lat.append(posibles[1][b])
        lon.append(posibles[2][b])
        despej.append(despejamiento)

    return [n_obs,d1,d2,elev,lat,lon,despej]

#Si solo hay un obstáculo
def unico_obs(wave_len,dis,d1,d2,despejamiento):
    #parámetros
    v=despejamiento*math.sqrt((2/wave_len)*((1/d1)+(1/d2)))
    radio = raw_input("¿Se conoce el radio del obstáculo? s o n: ")
    #si no se conoce el radio->filo de cuchillo
    if radio=="n":
        #las perdidas son igual a J(v)
        perdidas=calc_Jv(v)
    #si se conoce el radio
    elif radio=="s":
        #calcula J(v)
        J_v=calc_Jv(v)
        #se pide el radio pues es conocido
        radio = float(raw_input("Introduzca el valor del radio en metros: "))
        #se calcula T(m,n)
        T_mn=calc_Tmn(radio,wave_len,d1,d2,despejamiento)
        #se suman los valores para calcular las perdidas
        perdidas=J_v+T_mn
    else:
        #si no se introdujo ni s ni n
        perdidas="error"
    return perdidas

#En esta funcion se calcula J(v)
def calc_Jv(v):
    #Si es mayor a -0.78
    if v>=(-0.78):
        J_v=6.9+20*math.log10(math.sqrt((v-0.1)**2+1)+v-0.1)
    else:
        #en otro caso, se redondea y se deja el v introducido como un valor con dos
decimales
        v_mod="{0:.2f}".format(round(v,2))
        #Los valores de Cv y Sv se cogen del diccionario que se encuentra en el otro
archivo que relaciona v con los valores correspondientes
        J_v=-20*math.log10((math.sqrt((1-Cv_Sv[v_mod][0]-
Cv_Sv[v_mod][1])**2)+(Cv_Sv[v_mod][0]-Cv_Sv[v_mod][1])**2)/2)
    return J_v

#En esta función se calcula el valor de T(m,n) necesario para las perdidas
def calc_Tmn(r, wave_len,d1,d2,h):
    #Se calcula m
    m=(r*(d1+d2)/(d1*d2))/((math.pi*r)/wave_len)**(float(1)/float(3))
    #Se calcula n
    n=(h*((math.pi*r)/wave_len)**(float(2)/float(3)))/(r)

    #Se calcula dependiendo del valor de m y n
    if m*n<=4:
        T_mn=7.2*math.sqrt(m)-(2-12.5*n)*m+3.6*m**float(3/2)-0.8*m**2

```

```

else:
    T_mn=-6-20*math.log10(m*n)+7.2*math.sqrt(m)-(2-17*n)*m+3.6*m**float(3/2)-
0.8*m**2

return T_mn

#Funcion que calcula las perdidas si hay dos obstaculos
def dos_obs(wave_len,dis,d1,d2,despejamiento,elev,lat,lon,a_e,h_RX):
    #Se calcula cada una de las v
    v=despejamiento[0]*math.sqrt((2/wave_len)*((1/d1[0])+(1/d2[0])))
    #Perdidas del obstaculo 1 suponiendo filo de cuchillo
    L_1=calc_Jv(v)
    #calculamos la distancia del obstaculo 2 al 1
    dis_obs1=GetDistance(lat[0],lon[0],lat[1],lon[1],a_e)
    #punto en el que se encuentra el obstaculo 2 frente a la linea recta que une el
    obstaculo 1 con el RX
    y_i=(h_RX-despejamiento[0])/d2[0]*(dis_obs1)+despejamiento[0]
    #Nuevo despejamiento entre ese punto y la elevacion del obstaculo 2
    h_2_nueva=elev[1]-y_i
    v2=h_2_nueva*math.sqrt((2/wave_len)*((1/dis_obs1)+(1/d2[1])))
    #Perdidas producidas por el obstaculo 2 suponiendo filo de duchillo
    L_2=calc_Jv(v2)
    #Calculo del termino de correccion
    T_c=calc_Tc(wave_len,d1,d2,dis_obs1,despejamiento)

    #Si se conoce el radio, habrá que calcular Tmn y tenerlo en cuenta en el resultado
    final
    radio = raw_input("¿Se conoce el radio del obstáculo? s o n: ")
    if radio=="n":
        #Perdidas suponiendo filo de duchillo
        perdidas=L_1+L_2-T_c
    elif radio=="s":
        radio1 = float(raw_input("Introduzca el valor del radio del primer obstáculo
en metros: "))
        T_mn1=calc_Tmn(radio1,wave_len,d1[0],d2[0],despejamiento[0])
        radio2 = float(raw_input("Introduzca el valor del radio del segundo obstáculo
en metros: "))
        T_mn2=calc_Tmn(radio2,wave_len,dis_obs1,d2[1],h_2_nueva)
        #Perdidas teniendo en cuenta el radio
        perdidas=L_1+T_mn1+L_2+T_mn2-T_c
    else:
        perdidas="error"

return perdidas

#Funcion que calcula el termino de correccion
def calc_Tc(wave_len,d1,d2,dis_obs,despejamiento):
    despejamiento[0]=abs(despejamiento[0])
    despejamiento[1]=abs(despejamiento[1])
    #Variable p
    p=math.sqrt((2*(d1[0]+dis_obs+d2[1]))/(wave_len*(dis_obs+d2[1])*d1[0]))*despejamie
nto[0]
    #Variable q
    q=math.sqrt((2*(d1[0]+dis_obs+d2[1]))/(wave_len*(dis_obs+d1[0])*d2[1]))*despejamie
nto[1]
    #parametro alpha
    alpha=math.atan(math.sqrt((dis_obs*(d1[0]+dis_obs+d2[1]))/(d1[0]*d2[1])))
    #Calculo del termino con los datos calculado anteriormente
    T_c=(12-20*math.log10(2/(1-alpha/math.pi)))*(q/p)**(2*p)
    return T_c

```


apiTFG.py

```

# Copyright Google Inc. 2010 All Rights Reserved
import json
import urllib
import math

ELEVATION_BASE_URL = 'https://maps.google.com/maps/api/elevation/json'
CHART_BASE_URL = 'http://chart.googleapis.com/chart'

DISTANCE_BASE_URL=
'https://maps.googleapis.com/maps/api/distancematrix/json?units=metric&mode=walking&'

#esta es mi key con mi email.
key="key=key_privada_1"
key_dis="key=key_privada_2"

def getChart(chartData, chartDataScaling="-500,1500",
chartType="lc",chartLabel="Elevation in Meters",chartSize="1000x300",
chartColor="orange", **chart_args):
    chart_args.update({
        'cht': chartType,
        'chs': chartSize,
        'chl': chartLabel,
        'chco': chartColor,
        'chds': chartDataScaling,
        'chxt': 'x,y',
        'chxr': '1,-500,1500'
    })

    dataString = 't:' + ','.join(str(x) for x in chartData)
    chart_args['chd'] = dataString.strip(',')

    chartUrl = CHART_BASE_URL + '?' + urllib.urlencode(chart_args)

    print("")
    print("Elevation Chart URL:")
    print("")
    print chartUrl

#Funcion que analiza el terreno entre ambas antenas
def getElevation(path, samples, sensor="false", **elvt_n_args):
    elvt_n_args.update({
        'path': path,
        'samples': samples,
        'sensor': sensor
    })

    #URL base para llamar a la API de Google
    url = ELEVATION_BASE_URL + '?' + urllib.urlencode(elvt_n_args) + '&' +key
    #Respuesta de la API
    response = json.load(urllib.urlopen(url))

    # Create a dictionary for each results[] object
    elevationArray = []
    latitudArray=[]
    longitudArray=[]

    #Bucle donde se va guardando lo que se va a devolver, la elevacion y la latitud y
    longitud de cada uno de los puntos
    for resultset in response['results']:
        elevationArray.append(resultset['elevation'])
        latitudArray.append(resultset['location']['lat'])
        longitudArray.append(resultset['location']['lng'])

    return [elevationArray,latitudArray,longitudArray]

#Funcion que calcula la distancia entre dos puntos

```

```
def GetDistance(lat1, lon1, lat2, lon2,R):
    rad=math.pi/180
    dlat=lat2-lat1
    dlon=lon2-lon1
    a=(math.sin(rad*dlat/2))**2 +
math.cos(rad*lat1)*math.cos(rad*lat2)*(math.sin(rad*dlon/2))**2
    distancia=2*R*math.asin(math.sqrt(a))*1000
    return distancia
```

Cv_Sv.py

```
Cv_Sv=dict({"-10.00": [-0.4999, -0.4682],
"-9.99": [-0.4901, -0.4697],
"-9.98": [-0.4812, -0.4742],
"-9.97": [-0.4741, -0.4813],
"-9.96": [-0.4696, -0.4901],
"-9.95": [-0.4680, -0.5000],
"-9.94": [-0.4695, -0.5098],
"-9.93": [-0.4740, -0.5187],
"-9.92": [-0.4810, -0.5258],
"-9.91": [-0.4898, -0.5305],
"-9.90": [-0.4996, -0.5321],
"-9.89": [-0.5095, -0.5308],
"-9.88": [-0.5184, -0.5264],
"-9.87": [-0.5256, -0.5196],
"-9.86": [-0.5304, -0.5108],
"-9.85": [-0.5323, -0.5010],
"-9.84": [-0.5311, -0.4911],
"-9.83": [-0.5270, -0.4821],
"-9.82": [-0.5203, -0.4747],
"-9.81": [-0.5117, -0.4697],
"-9.80": [-0.5019, -0.4676],
"-9.79": [-0.4920, -0.4685],
"-9.78": [-0.4828, -0.4723],
"-9.77": [-0.4753, -0.4788],
"-9.76": [-0.4700, -0.4873],
"-9.75": [-0.4675, -0.4969],
"-9.74": [-0.4680, -0.5069],
"-9.73": [-0.4716, -0.5162],
"-9.72": [-0.4777, -0.5240],
"-9.71": [-0.4860, -0.5296],
"-9.70": [-0.4955, -0.5325],
"-9.69": [-0.5054, -0.5324],
"-9.68": [-0.5149, -0.5293],
"-9.67": [-0.5230, -0.5235],
"-9.66": [-0.5290, -0.5156],
"-9.65": [-0.5324, -0.5062],
"-9.64": [-0.5328, -0.4962],
"-9.63": [-0.5302, -0.4866],
"-9.62": [-0.5249, -0.4782],
"-9.61": [-0.5173, -0.4718],
"-9.60": [-0.5081, -0.4679],
"-9.59": [-0.4982, -0.4669],
"-9.58": [-0.4885, -0.4688],
"-9.57": [-0.4797, -0.4736],
"-9.56": [-0.4728, -0.4808],
"-9.55": [-0.4683, -0.4897],
"-9.54": [-0.4666, -0.4995],
"-9.53": [-0.4679, -0.5094],
"-9.52": [-0.4721, -0.5184],
"-9.51": [-0.4788, -0.5259],
"-9.50": [-0.4873, -0.5310],
"-9.49": [-0.4970, -0.5334],
"-9.48": [-0.5069, -0.5329],
"-9.47": [-0.5163, -0.5294],
"-9.46": [-0.5242, -0.5234],
"-9.45": [-0.5300, -0.5153],
"-9.44": [-0.5332, -0.5059],
"-9.43": [-0.5335, -0.4959],
"-9.42": [-0.5309, -0.4863],
"-9.41": [-0.5256, -0.4779],
"-9.40": [-0.5180, -0.4713],
"-9.39": [-0.5089, -0.4673],
"-9.38": [-0.4990, -0.4661],
"-9.37": [-0.4892, -0.4678],
"-9.36": [-0.4803, -0.4723],
"-9.35": [-0.4731, -0.4791],
"-9.34": [-0.4682, -0.4878],
```

"-9.33": [-0.4660, -0.4975],
"-9.32": [-0.4667, -0.5074],
"-9.31": [-0.4702, -0.5168],
"-9.30": [-0.4763, -0.5247],
"-9.29": [-0.4844, -0.5305],
"-9.28": [-0.4938, -0.5337],
"-9.27": [-0.5037, -0.5341],
"-9.26": [-0.5134, -0.5317],
"-9.25": [-0.5219, -0.5265],
"-9.24": [-0.5286, -0.5191],
"-9.23": [-0.5330, -0.5102],
"-9.22": [-0.5345, -0.5003],
"-9.21": [-0.5332, -0.4904],
"-9.20": [-0.5291, -0.4814],
"-9.19": [-0.5226, -0.4738],
"-9.18": [-0.5143, -0.4684],
"-9.17": [-0.5047, -0.4656],
"-9.16": [-0.4947, -0.4657],
"-9.15": [-0.4852, -0.4685],
"-9.14": [-0.4769, -0.4740],
"-9.13": [-0.4704, -0.4816],
"-9.12": [-0.4664, -0.4907],
"-9.11": [-0.4651, -0.5006],
"-9.10": [-0.4666, -0.5104],
"-9.09": [-0.4709, -0.5194],
"-9.08": [-0.4775, -0.5269],
"-9.07": [-0.4859, -0.5322],
"-9.06": [-0.4955, -0.5348],
"-9.05": [-0.5055, -0.5347],
"-9.04": [-0.5150, -0.5318],
"-9.03": [-0.5234, -0.5264],
"-9.02": [-0.5299, -0.5188],
"-9.01": [-0.5340, -0.5097],
"-9.00": [-0.5354, -0.4999],
"-8.99": [-0.5340, -0.4900],
"-8.98": [-0.5299, -0.4809],
"-8.97": [-0.5234, -0.4733],
"-8.96": [-0.5151, -0.4678],
"-8.95": [-0.5056, -0.4649],
"-8.94": [-0.4956, -0.4647],
"-8.93": [-0.4860, -0.4672],
"-8.92": [-0.4774, -0.4724],
"-8.91": [-0.4706, -0.4797],
"-8.90": [-0.4661, -0.4886],
"-8.89": [-0.4642, -0.4983],
"-8.88": [-0.4651, -0.5083],
"-8.87": [-0.4687, -0.5176],
"-8.86": [-0.4747, -0.5255],
"-8.85": [-0.4827, -0.5315],
"-8.84": [-0.4920, -0.5351],
"-8.83": [-0.5019, -0.5360],
"-8.82": [-0.5117, -0.5341],
"-8.81": [-0.5206, -0.5297],
"-8.80": [-0.5280, -0.5229],
"-8.79": [-0.5332, -0.5145],
"-8.78": [-0.5359, -0.5049],
"-8.77": [-0.5359, -0.4949],
"-8.76": [-0.5332, -0.4853],
"-8.75": [-0.5280, -0.4768],
"-8.74": [-0.5207, -0.4700],
"-8.73": [-0.5118, -0.4655],
"-8.72": [-0.5020, -0.4636],
"-8.71": [-0.4921, -0.4643],
"-8.70": [-0.4827, -0.4677],
"-8.69": [-0.4747, -0.4736],
"-8.68": [-0.4684, -0.4813],
"-8.67": [-0.4645, -0.4905],
"-8.66": [-0.4632, -0.5004],
"-8.65": [-0.4647, -0.5103],
"-8.64": [-0.4687, -0.5194],

```
"-8.63": [-0.4750, -0.5271],
"-8.62": [-0.4831, -0.5329],
"-8.61": [-0.4925, -0.5362],
"-8.60": [-0.5025, -0.5369],
"-8.59": [-0.5123, -0.5350],
"-8.58": [-0.5212, -0.5305],
"-8.57": [-0.5285, -0.5238],
"-8.56": [-0.5339, -0.5153],
"-8.55": [-0.5368, -0.5058],
"-8.54": [-0.5370, -0.4958],
"-8.53": [-0.5347, -0.4862],
"-8.52": [-0.5298, -0.4775],
"-8.51": [-0.5228, -0.4703],
"-8.50": [-0.5142, -0.4653],
"-8.49": [-0.5045, -0.4628],
"-8.48": [-0.4946, -0.4629],
"-8.47": [-0.4850, -0.4656],
"-8.46": [-0.4764, -0.4707],
"-8.45": [-0.4695, -0.4779],
"-8.44": [-0.4647, -0.4866],
"-8.43": [-0.4624, -0.4963],
"-8.42": [-0.4627, -0.5063],
"-8.41": [-0.4656, -0.5158],
"-8.40": [-0.4709, -0.5243],
"-8.39": [-0.4782, -0.5311],
"-8.38": [-0.4870, -0.5357],
"-8.37": [-0.4968, -0.5379],
"-8.36": [-0.5067, -0.5375],
"-8.35": [-0.5163, -0.5345],
"-8.34": [-0.5247, -0.5291],
"-8.33": [-0.5314, -0.5218],
"-8.32": [-0.5360, -0.5129],
"-8.31": [-0.5382, -0.5032],
"-8.30": [-0.5377, -0.4932],
"-8.29": [-0.5348, -0.4837],
"-8.28": [-0.5294, -0.4753],
"-8.27": [-0.5221, -0.4685],
"-8.26": [-0.5133, -0.4638],
"-8.25": [-0.5036, -0.4616],
"-8.24": [-0.4936, -0.4619],
"-8.23": [-0.4841, -0.4648],
"-8.22": [-0.4756, -0.4700],
"-8.21": [-0.4687, -0.4772],
"-8.20": [-0.4638, -0.4859],
"-8.19": [-0.4614, -0.4955],
"-8.18": [-0.4615, -0.5055],
"-8.17": [-0.4641, -0.5151],
"-8.16": [-0.4691, -0.5238],
"-8.15": [-0.4761, -0.5309],
"-8.14": [-0.4847, -0.5360],
"-8.13": [-0.4943, -0.5387],
"-8.12": [-0.5042, -0.5390],
"-8.11": [-0.5139, -0.5367],
"-8.10": [-0.5228, -0.5320],
"-8.09": [-0.5301, -0.5253],
"-8.08": [-0.5356, -0.5170],
"-8.07": [-0.5387, -0.5075],
"-8.06": [-0.5394, -0.4976],
"-8.05": [-0.5376, -0.4877],
"-8.04": [-0.5334, -0.4787],
"-8.03": [-0.5270, -0.4710],
"-8.02": [-0.5190, -0.4651],
"-8.01": [-0.5097, -0.4615],
"-8.00": [-0.4998, -0.4602],
"-7.99": [-0.4899, -0.4615],
"-7.98": [-0.4806, -0.4651],
"-7.97": [-0.4726, -0.4710],
"-7.96": [-0.4662, -0.4787],
"-7.95": [-0.4619, -0.4877],
"-7.94": [-0.4600, -0.4975],
```

"-7.93": [-0.4606, -0.5074],
"-7.92": [-0.4636, -0.5169],
"-7.91": [-0.4688, -0.5254],
"-7.90": [-0.4760, -0.5323],
"-7.89": [-0.4846, -0.5373],
"-7.88": [-0.4942, -0.5400],
"-7.87": [-0.5042, -0.5402],
"-7.86": [-0.5139, -0.5380],
"-7.85": [-0.5228, -0.5335],
"-7.84": [-0.5304, -0.5270],
"-7.83": [-0.5361, -0.5188],
"-7.82": [-0.5396, -0.5094],
"-7.81": [-0.5408, -0.4995],
"-7.80": [-0.5395, -0.4896],
"-7.79": [-0.5358, -0.4804],
"-7.78": [-0.5300, -0.4722],
"-7.77": [-0.5225, -0.4657],
"-7.76": [-0.5135, -0.4613],
"-7.75": [-0.5038, -0.4591],
"-7.74": [-0.4938, -0.4593],
"-7.73": [-0.4842, -0.4620],
"-7.72": [-0.4755, -0.4668],
"-7.71": [-0.4682, -0.4737],
"-7.70": [-0.4628, -0.4820],
"-7.69": [-0.4595, -0.4914],
"-7.68": [-0.4586, -0.5014],
"-7.67": [-0.4601, -0.5112],
"-7.66": [-0.4638, -0.5205],
"-7.65": [-0.4697, -0.5285],
"-7.64": [-0.4773, -0.5350],
"-7.63": [-0.4863, -0.5394],
"-7.62": [-0.4960, -0.5416],
"-7.61": [-0.5060, -0.5414],
"-7.60": [-0.5156, -0.5389],
"-7.59": [-0.5244, -0.5341],
"-7.58": [-0.5318, -0.5274],
"-7.57": [-0.5374, -0.5192],
"-7.56": [-0.5409, -0.5098],
"-7.55": [-0.5422, -0.4999],
"-7.54": [-0.5410, -0.4900],
"-7.53": [-0.5376, -0.4807],
"-7.52": [-0.5320, -0.4724],
"-7.51": [-0.5247, -0.4656],
"-7.50": [-0.5160, -0.4607],
"-7.49": [-0.5064, -0.4580],
"-7.48": [-0.4964, -0.4576],
"-7.47": [-0.4867, -0.4595],
"-7.46": [-0.4776, -0.4637],
"-7.45": [-0.4697, -0.4698],
"-7.44": [-0.4635, -0.4777],
"-7.43": [-0.4593, -0.4867],
"-7.42": [-0.4573, -0.4965],
"-7.41": [-0.4575, -0.5064],
"-7.40": [-0.4601, -0.5161],
"-7.39": [-0.4648, -0.5249],
"-7.38": [-0.4714, -0.5323],
"-7.37": [-0.4796, -0.5381],
"-7.36": [-0.4889, -0.5418],
"-7.35": [-0.4987, -0.5433],
"-7.34": [-0.5087, -0.5425],
"-7.33": [-0.5182, -0.5394],
"-7.32": [-0.5267, -0.5343],
"-7.31": [-0.5339, -0.5273],
"-7.30": [-0.5393, -0.5189],
"-7.29": [-0.5426, -0.5095],
"-7.28": [-0.5437, -0.4996],
"-7.27": [-0.5426, -0.4897],
"-7.26": [-0.5392, -0.4803],
"-7.25": [-0.5338, -0.4719],
"-7.24": [-0.5266, -0.4650],

```
"-7.23": [-0.5181, -0.4599],
"-7.22": [-0.5086, -0.4568],
"-7.21": [-0.4986, -0.4559],
"-7.20": [-0.4887, -0.4573],
"-7.19": [-0.4794, -0.4608],
"-7.18": [-0.4711, -0.4664],
"-7.17": [-0.4643, -0.4736],
"-7.16": [-0.4592, -0.4822],
"-7.15": [-0.4563, -0.4918],
"-7.14": [-0.4555, -0.5017],
"-7.13": [-0.4569, -0.5116],
"-7.12": [-0.4605, -0.5209],
"-7.11": [-0.4660, -0.5292],
"-7.10": [-0.4733, -0.5360],
"-7.09": [-0.4819, -0.5411],
"-7.08": [-0.4914, -0.5441],
"-7.07": [-0.5014, -0.5450],
"-7.06": [-0.5112, -0.5437],
"-7.05": [-0.5206, -0.5402],
"-7.04": [-0.5290, -0.5347],
"-7.03": [-0.5359, -0.5276],
"-7.02": [-0.5411, -0.5191],
"-7.01": [-0.5444, -0.5096],
"-7.00": [-0.5455, -0.4997],
"-6.99": [-0.5444, -0.4898],
"-6.98": [-0.5411, -0.4803],
"-6.97": [-0.5359, -0.4718],
"-6.96": [-0.5290, -0.4646],
"-6.95": [-0.5207, -0.4591],
"-6.94": [-0.5114, -0.4556],
"-6.93": [-0.5015, -0.4541],
"-6.92": [-0.4915, -0.4548],
"-6.91": [-0.4820, -0.4576],
"-6.90": [-0.4732, -0.4624],
"-6.89": [-0.4657, -0.4690],
"-6.88": [-0.4598, -0.4771],
"-6.87": [-0.4558, -0.4862],
"-6.86": [-0.4538, -0.4960],
"-6.85": [-0.4539, -0.5060],
"-6.84": [-0.4562, -0.5157],
"-6.83": [-0.4605, -0.5247],
"-6.82": [-0.4666, -0.5326],
"-6.81": [-0.4742, -0.5390],
"-6.80": [-0.4831, -0.5436],
"-6.79": [-0.4927, -0.5463],
"-6.78": [-0.5027, -0.5469],
"-6.77": [-0.5125, -0.5453],
"-6.76": [-0.5218, -0.5417],
"-6.75": [-0.5302, -0.5362],
"-6.74": [-0.5372, -0.5291],
"-6.73": [-0.5425, -0.5207],
"-6.72": [-0.5460, -0.5113],
"-6.71": [-0.5474, -0.5015],
"-6.70": [-0.5467, -0.4915],
"-6.69": [-0.5440, -0.4819],
"-6.68": [-0.5393, -0.4731],
"-6.67": [-0.5329, -0.4654],
"-6.66": [-0.5250, -0.4593],
"-6.65": [-0.5161, -0.4549],
"-6.64": [-0.5064, -0.4525],
"-6.63": [-0.4964, -0.4521],
"-6.62": [-0.4866, -0.4538],
"-6.61": [-0.4773, -0.4575],
"-6.60": [-0.4690, -0.4631],
"-6.59": [-0.4620, -0.4702],
"-6.58": [-0.4566, -0.4786],
"-6.57": [-0.4531, -0.4879],
"-6.56": [-0.4515, -0.4978],
"-6.55": [-0.4520, -0.5078],
"-6.54": [-0.4546, -0.5174],
```

"-6.53": [-0.4590, -0.5264],
"-6.52": [-0.4652, -0.5342],
"-6.51": [-0.4728, -0.5406],
"-6.50": [-0.4816, -0.5454],
"-6.49": [-0.4912, -0.5482],
"-6.48": [-0.5011, -0.5491],
"-6.47": [-0.5110, -0.5479],
"-6.46": [-0.5205, -0.5448],
"-6.45": [-0.5292, -0.5398],
"-6.44": [-0.5366, -0.5332],
"-6.43": [-0.5426, -0.5252],
"-6.42": [-0.5469, -0.5162],
"-6.41": [-0.5492, -0.5065],
"-6.40": [-0.5496, -0.4965],
"-6.39": [-0.5480, -0.4866],
"-6.38": [-0.5444, -0.4773],
"-6.37": [-0.5391, -0.4689],
"-6.36": [-0.5322, -0.4617],
"-6.35": [-0.5240, -0.4560],
"-6.34": [-0.5148, -0.4520],
"-6.33": [-0.5050, -0.4500],
"-6.32": [-0.4951, -0.4499],
"-6.31": [-0.4852, -0.4518],
"-6.30": [-0.4760, -0.4555],
"-6.29": [-0.4677, -0.4611],
"-6.28": [-0.4606, -0.4681],
"-6.27": [-0.4551, -0.4764],
"-6.26": [-0.4512, -0.4856],
"-6.25": [-0.4493, -0.4954],
"-6.24": [-0.4493, -0.5054],
"-6.23": [-0.4512, -0.5152],
"-6.22": [-0.4550, -0.5244],
"-6.21": [-0.4606, -0.5327],
"-6.20": [-0.4676, -0.5398],
"-6.19": [-0.4759, -0.5454],
"-6.18": [-0.4851, -0.5493],
"-6.17": [-0.4949, -0.5513],
"-6.16": [-0.5048, -0.5514],
"-6.15": [-0.5147, -0.5496],
"-6.14": [-0.5239, -0.5460],
"-6.13": [-0.5324, -0.5406],
"-6.12": [-0.5396, -0.5337],
"-6.11": [-0.5454, -0.5256],
"-6.10": [-0.5495, -0.5165],
"-6.09": [-0.5518, -0.5068],
"-6.08": [-0.5522, -0.4968],
"-6.07": [-0.5508, -0.4869],
"-6.06": [-0.5475, -0.4775],
"-6.05": [-0.5424, -0.4689],
"-6.04": [-0.5358, -0.4614],
"-6.03": [-0.5280, -0.4552],
"-6.02": [-0.5191, -0.4507],
"-6.01": [-0.5095, -0.4479],
"-6.00": [-0.4995, -0.4470],
"-5.99": [-0.4896, -0.4479],
"-5.98": [-0.4800, -0.4507],
"-5.97": [-0.4711, -0.4552],
"-5.96": [-0.4632, -0.4613],
"-5.95": [-0.4566, -0.4688],
"-5.94": [-0.4514, -0.4774],
"-5.93": [-0.4480, -0.4867],
"-5.92": [-0.4464, -0.4966],
"-5.91": [-0.4466, -0.5066],
"-5.90": [-0.4486, -0.5163],
"-5.89": [-0.4524, -0.5256],
"-5.88": [-0.4578, -0.5339],
"-5.87": [-0.4647, -0.5412],
"-5.86": [-0.4728, -0.5470],
"-5.85": [-0.4818, -0.5513],
"-5.84": [-0.4915, -0.5538],

```
"-5.83": [-0.5015, -0.5546],
"-5.82": [-0.5114, -0.5535],
"-5.81": [-0.5210, -0.5506],
"-5.80": [-0.5298, -0.5460],
"-5.79": [-0.5377, -0.5400],
"-5.78": [-0.5444, -0.5325],
"-5.77": [-0.5497, -0.5240],
"-5.76": [-0.5533, -0.5147],
"-5.75": [-0.5551, -0.5049],
"-5.74": [-0.5552, -0.4949],
"-5.73": [-0.5535, -0.4851],
"-5.72": [-0.5500, -0.4757],
"-5.71": [-0.5450, -0.4671],
"-5.70": [-0.5385, -0.4595],
"-5.69": [-0.5307, -0.4532],
"-5.68": [-0.5219, -0.4484],
"-5.67": [-0.5125, -0.4453],
"-5.66": [-0.5026, -0.4438],
"-5.65": [-0.4926, -0.4442],
"-5.64": [-0.4828, -0.4462],
"-5.63": [-0.4736, -0.4500],
"-5.62": [-0.4652, -0.4554],
"-5.61": [-0.4578, -0.4621],
"-5.60": [-0.4517, -0.4700],
"-5.59": [-0.4471, -0.4789],
"-5.58": [-0.4442, -0.4884],
"-5.57": [-0.4429, -0.4984],
"-5.56": [-0.4434, -0.5083],
"-5.55": [-0.4456, -0.5181],
"-5.54": [-0.4494, -0.5273],
"-5.53": [-0.4548, -0.5357],
"-5.52": [-0.4616, -0.5430],
"-5.51": [-0.4696, -0.5491],
"-5.50": [-0.4784, -0.5537],
"-5.49": [-0.4879, -0.5567],
"-5.48": [-0.4978, -0.5580],
"-5.47": [-0.5078, -0.5576],
"-5.46": [-0.5176, -0.5556],
"-5.45": [-0.5269, -0.5518],
"-5.44": [-0.5354, -0.5466],
"-5.43": [-0.5428, -0.5400],
"-5.42": [-0.5491, -0.5322],
"-5.41": [-0.5539, -0.5235],
"-5.40": [-0.5572, -0.5140],
"-5.39": [-0.5589, -0.5042],
"-5.38": [-0.5589, -0.4942],
"-5.37": [-0.5572, -0.4844],
"-5.36": [-0.5538, -0.4749],
"-5.35": [-0.5490, -0.4662],
"-5.34": [-0.5427, -0.4584],
"-5.33": [-0.5352, -0.4518],
"-5.32": [-0.5268, -0.4465],
"-5.31": [-0.5175, -0.4427],
"-5.30": [-0.5078, -0.4405],
"-5.29": [-0.4978, -0.4399],
"-5.28": [-0.4879, -0.4410],
"-5.27": [-0.4783, -0.4437],
"-5.26": [-0.4692, -0.4479],
"-5.25": [-0.4610, -0.4536],
"-5.24": [-0.4538, -0.4605],
"-5.23": [-0.4479, -0.4686],
"-5.22": [-0.4434, -0.4775],
"-5.21": [-0.4403, -0.4870],
"-5.20": [-0.4389, -0.4969],
"-5.19": [-0.4391, -0.5069],
"-5.18": [-0.4409, -0.5167],
"-5.17": [-0.4443, -0.5261],
"-5.16": [-0.4491, -0.5348],
"-5.15": [-0.4553, -0.5427],
"-5.14": [-0.4627, -0.5494],
```

"-5.13": [-0.4710, -0.5549],
"-5.12": [-0.4802, -0.5589],
"-5.11": [-0.4898, -0.5614],
"-5.10": [-0.4998, -0.5624],
"-5.09": [-0.5098, -0.5617],
"-5.08": [-0.5195, -0.5595],
"-5.07": [-0.5288, -0.5558],
"-5.06": [-0.5373, -0.5506],
"-5.05": [-0.5449, -0.5442],
"-5.04": [-0.5514, -0.5366],
"-5.03": [-0.5567, -0.5281],
"-5.02": [-0.5605, -0.5189],
"-5.01": [-0.5628, -0.5092],
"-5.00": [-0.5636, -0.4992],
"-4.99": [-0.5628, -0.4892],
"-4.98": [-0.5605, -0.4795],
"-4.97": [-0.5567, -0.4703],
"-4.96": [-0.5515, -0.4618],
"-4.95": [-0.5450, -0.4541],
"-4.94": [-0.5375, -0.4476],
"-4.93": [-0.5290, -0.4423],
"-4.92": [-0.5198, -0.4384],
"-4.91": [-0.5101, -0.4360],
"-4.90": [-0.5002, -0.4351],
"-4.89": [-0.4902, -0.4357],
"-4.88": [-0.4804, -0.4378],
"-4.87": [-0.4711, -0.4414],
"-4.86": [-0.4624, -0.4464],
"-4.85": [-0.4546, -0.4526],
"-4.84": [-0.4479, -0.4600],
"-4.83": [-0.4423, -0.4682],
"-4.82": [-0.4380, -0.4773],
"-4.81": [-0.4352, -0.4869],
"-4.80": [-0.4338, -0.4968],
"-4.79": [-0.4339, -0.5067],
"-4.78": [-0.4355, -0.5166],
"-4.77": [-0.4386, -0.5261],
"-4.76": [-0.4431, -0.5350],
"-4.75": [-0.4488, -0.5432],
"-4.74": [-0.4557, -0.5504],
"-4.73": [-0.4636, -0.5566],
"-4.72": [-0.4723, -0.5615],
"-4.71": [-0.4817, -0.5650],
"-4.70": [-0.4914, -0.5671],
"-4.69": [-0.5014, -0.5678],
"-4.68": [-0.5114, -0.5670],
"-4.67": [-0.5211, -0.5648],
"-4.66": [-0.5304, -0.5611],
"-4.65": [-0.5391, -0.5562],
"-4.64": [-0.5469, -0.5500],
"-4.63": [-0.5538, -0.5427],
"-4.62": [-0.5595, -0.5346],
"-4.61": [-0.5641, -0.5257],
"-4.60": [-0.5672, -0.5162],
"-4.59": [-0.5690, -0.5064],
"-4.58": [-0.5694, -0.4964],
"-4.57": [-0.5683, -0.4864],
"-4.56": [-0.5658, -0.4768],
"-4.55": [-0.5619, -0.4675],
"-4.54": [-0.5568, -0.4590],
"-4.53": [-0.5505, -0.4512],
"-4.52": [-0.5432, -0.4444],
"-4.51": [-0.5350, -0.4387],
"-4.50": [-0.5260, -0.4343],
"-4.49": [-0.5165, -0.4311],
"-4.48": [-0.5067, -0.4293],
"-4.47": [-0.4967, -0.4289],
"-4.46": [-0.4868, -0.4299],
"-4.45": [-0.4771, -0.4323],
"-4.44": [-0.4678, -0.4360],

```

"-4.43": [-0.4591, -0.4410],
"-4.42": [-0.4512, -0.4471],
"-4.41": [-0.4443, -0.4542],
"-4.40": [-0.4383, -0.4623],
"-4.39": [-0.4336, -0.4711],
"-4.38": [-0.4301, -0.4804],
"-4.37": [-0.4279, -0.4902],
"-4.36": [-0.4270, -0.5001],
"-4.35": [-0.4276, -0.5101],
"-4.34": [-0.4295, -0.5199],
"-4.33": [-0.4327, -0.5294],
"-4.32": [-0.4371, -0.5383],
"-4.31": [-0.4428, -0.5466],
"-4.30": [-0.4494, -0.5540],
"-4.29": [-0.4571, -0.5604],
"-4.28": [-0.4655, -0.5658],
"-4.27": [-0.4746, -0.5700],
"-4.26": [-0.4841, -0.5730],
"-4.25": [-0.4940, -0.5746],
"-4.24": [-0.5040, -0.5749],
"-4.23": [-0.5139, -0.5739],
"-4.22": [-0.5236, -0.5716],
"-4.21": [-0.5329, -0.5680],
"-4.20": [-0.5417, -0.5632],
"-4.19": [-0.5498, -0.5573],
"-4.18": [-0.5570, -0.5504],
"-4.17": [-0.5633, -0.5426],
"-4.16": [-0.5684, -0.5341],
"-4.15": [-0.5725, -0.5249],
"-4.14": [-0.5753, -0.5153],
"-4.13": [-0.5768, -0.5054],
"-4.12": [-0.5771, -0.4955],
"-4.11": [-0.5760, -0.4855],
"-4.10": [-0.5737, -0.4758],
"-4.09": [-0.5701, -0.4665],
"-4.08": [-0.5654, -0.4576],
"-4.07": [-0.5596, -0.4495],
"-4.06": [-0.5528, -0.4422],
"-4.05": [-0.5452, -0.4358],
"-4.04": [-0.5368, -0.4304],
"-4.03": [-0.5277, -0.4261],
"-4.02": [-0.5182, -0.4230],
"-4.01": [-0.5084, -0.4211],
"-4.00": [-0.4984, -0.4205],
"-3.99": [-0.4885, -0.4211],
"-3.98": [-0.4786, -0.4230],
"-3.97": [-0.4691, -0.4261],
"-3.96": [-0.4601, -0.4303],
"-3.95": [-0.4516, -0.4357],
"-3.94": [-0.4439, -0.4420],
"-3.93": [-0.4370, -0.4492],
"-3.92": [-0.4311, -0.4573],
"-3.91": [-0.4261, -0.4660],
"-3.90": [-0.4223, -0.4752],
"-3.89": [-0.4197, -0.4848],
"-3.88": [-0.4182, -0.4947],
"-3.87": [-0.4180, -0.5047],
"-3.86": [-0.4189, -0.5147],
"-3.85": [-0.4211, -0.5244],
"-3.84": [-0.4244, -0.5338],
"-3.83": [-0.4289, -0.5428],
"-3.82": [-0.4343, -0.5512],
"-3.81": [-0.4408, -0.5588],
"-3.80": [-0.4481, -0.5656],
"-3.79": [-0.4562, -0.5715],
"-3.78": [-0.4649, -0.5764],
"-3.77": [-0.4741, -0.5803],
"-3.76": [-0.4837, -0.5830],
"-3.75": [-0.4936, -0.5845],
"-3.74": [-0.5036, -0.5849],

```

"-3.73": [-0.5135, -0.5841],
"-3.72": [-0.5233, -0.5822],
"-3.71": [-0.5329, -0.5791],
"-3.70": [-0.5419, -0.5750],
"-3.69": [-0.5505, -0.5698],
"-3.68": [-0.5584, -0.5637],
"-3.67": [-0.5655, -0.5567],
"-3.66": [-0.5718, -0.5489],
"-3.65": [-0.5771, -0.5404],
"-3.64": [-0.5815, -0.5314],
"-3.63": [-0.5848, -0.5220],
"-3.62": [-0.5869, -0.5122],
"-3.61": [-0.5880, -0.5023],
"-3.60": [-0.5880, -0.4923],
"-3.59": [-0.5868, -0.4824],
"-3.58": [-0.5845, -0.4727],
"-3.57": [-0.5811, -0.4633],
"-3.56": [-0.5767, -0.4543],
"-3.55": [-0.5713, -0.4459],
"-3.54": [-0.5650, -0.4381],
"-3.53": [-0.5579, -0.4311],
"-3.52": [-0.5501, -0.4249],
"-3.51": [-0.5416, -0.4196],
"-3.50": [-0.5326, -0.4152],
"-3.49": [-0.5231, -0.4119],
"-3.48": [-0.5134, -0.4097],
"-3.47": [-0.5035, -0.4085],
"-3.46": [-0.4935, -0.4084],
"-3.45": [-0.4835, -0.4094],
"-3.44": [-0.4738, -0.4114],
"-3.43": [-0.4643, -0.4145],
"-3.42": [-0.4551, -0.4186],
"-3.41": [-0.4465, -0.4237],
"-3.40": [-0.4385, -0.4296],
"-3.39": [-0.4311, -0.4364],
"-3.38": [-0.4246, -0.4439],
"-3.37": [-0.4188, -0.4521],
"-3.36": [-0.4139, -0.4608],
"-3.35": [-0.4100, -0.4700],
"-3.34": [-0.4071, -0.4796],
"-3.33": [-0.4052, -0.4894],
"-3.32": [-0.4043, -0.4994],
"-3.31": [-0.4045, -0.5094],
"-3.30": [-0.4057, -0.5193],
"-3.29": [-0.4079, -0.5290],
"-3.28": [-0.4111, -0.5385],
"-3.27": [-0.4153, -0.5476],
"-3.26": [-0.4204, -0.5562],
"-3.25": [-0.4263, -0.5642],
"-3.24": [-0.4331, -0.5716],
"-3.23": [-0.4405, -0.5783],
"-3.22": [-0.4486, -0.5842],
"-3.21": [-0.4572, -0.5892],
"-3.20": [-0.4663, -0.5933],
"-3.19": [-0.4758, -0.5966],
"-3.18": [-0.4855, -0.5988],
"-3.17": [-0.4954, -0.6001],
"-3.16": [-0.5054, -0.6003],
"-3.15": [-0.5154, -0.5996],
"-3.14": [-0.5253, -0.5979],
"-3.13": [-0.5349, -0.5952],
"-3.12": [-0.5442, -0.5917],
"-3.11": [-0.5531, -0.5872],
"-3.10": [-0.5616, -0.5818],
"-3.09": [-0.5695, -0.5757],
"-3.08": [-0.5767, -0.5688],
"-3.07": [-0.5833, -0.5613],
"-3.06": [-0.5891, -0.5531],
"-3.05": [-0.5941, -0.5445],
"-3.04": [-0.5982, -0.5354],

```
"-3.03": [-0.6015, -0.5259],
"-3.02": [-0.6038, -0.5162],
"-3.01": [-0.6052, -0.5063],
"-3.00": [-0.6057, -0.4963],
"-2.99": [-0.6053, -0.4863],
"-2.98": [-0.6038, -0.4764],
"-2.97": [-0.6015, -0.4667],
"-2.96": [-0.5983, -0.4572],
"-2.95": [-0.5942, -0.4481],
"-2.94": [-0.5893, -0.4394],
"-2.93": [-0.5836, -0.4312],
"-2.92": [-0.5772, -0.4235],
"-2.91": [-0.5701, -0.4165],
"-2.90": [-0.5624, -0.4101],
"-2.89": [-0.5541, -0.4045],
"-2.88": [-0.5454, -0.3996],
"-2.87": [-0.5362, -0.3956],
"-2.86": [-0.5268, -0.3924],
"-2.85": [-0.5171, -0.3900],
"-2.84": [-0.5072, -0.3886],
"-2.83": [-0.4972, -0.3880],
"-2.82": [-0.4872, -0.3883],
"-2.81": [-0.4773, -0.3895],
"-2.80": [-0.4675, -0.3915],
"-2.79": [-0.4579, -0.3944],
"-2.78": [-0.4487, -0.3982],
"-2.77": [-0.4397, -0.4027],
"-2.76": [-0.4313, -0.4080],
"-2.75": [-0.4233, -0.4140],
"-2.74": [-0.4158, -0.4207],
"-2.73": [-0.4090, -0.4279],
"-2.72": [-0.4028, -0.4358],
"-2.71": [-0.3973, -0.4441],
"-2.70": [-0.3925, -0.4529],
"-2.69": [-0.3885, -0.4621],
"-2.68": [-0.3853, -0.4715],
"-2.67": [-0.3828, -0.4812],
"-2.66": [-0.3812, -0.4911],
"-2.65": [-0.3805, -0.5011],
"-2.64": [-0.3805, -0.5111],
"-2.63": [-0.3814, -0.5210],
"-2.62": [-0.3831, -0.5309],
"-2.61": [-0.3856, -0.5406],
"-2.60": [-0.3889, -0.5500],
"-2.59": [-0.3930, -0.5591],
"-2.58": [-0.3978, -0.5679],
"-2.57": [-0.4033, -0.5763],
"-2.56": [-0.4094, -0.5842],
"-2.55": [-0.4161, -0.5915],
"-2.54": [-0.4235, -0.5983],
"-2.53": [-0.4313, -0.6045],
"-2.52": [-0.4396, -0.6101],
"-2.51": [-0.4483, -0.6150],
"-2.50": [-0.4574, -0.6192],
"-2.49": [-0.4668, -0.6226],
"-2.48": [-0.4764, -0.6254],
"-2.47": [-0.4862, -0.6273],
"-2.46": [-0.4961, -0.6285],
"-2.45": [-0.5061, -0.6289],
"-2.44": [-0.5161, -0.6286],
"-2.43": [-0.5261, -0.6275],
"-2.42": [-0.5359, -0.6256],
"-2.41": [-0.5455, -0.6230],
"-2.40": [-0.5550, -0.6197],
"-2.39": [-0.5641, -0.6157],
"-2.38": [-0.5729, -0.6110],
"-2.37": [-0.5814, -0.6056],
"-2.36": [-0.5894, -0.5996],
"-2.35": [-0.5970, -0.5931],
"-2.34": [-0.6040, -0.5860],
```

"-2.33": [-0.6105, -0.5784],
"-2.32": [-0.6165, -0.5704],
"-2.31": [-0.6218, -0.5620],
"-2.30": [-0.6266, -0.5532],
"-2.29": [-0.6306, -0.5440],
"-2.28": [-0.6340, -0.5346],
"-2.27": [-0.6368, -0.5250],
"-2.26": [-0.6388, -0.5152],
"-2.25": [-0.6401, -0.5053],
"-2.24": [-0.6408, -0.4953],
"-2.23": [-0.6407, -0.4853],
"-2.22": [-0.6399, -0.4754],
"-2.21": [-0.6384, -0.4655],
"-2.20": [-0.6363, -0.4557],
"-2.19": [-0.6335, -0.4461],
"-2.18": [-0.6300, -0.4367],
"-2.17": [-0.6259, -0.4276],
"-2.16": [-0.6212, -0.4188],
"-2.15": [-0.6159, -0.4103],
"-2.14": [-0.6100, -0.4022],
"-2.13": [-0.6036, -0.3945],
"-2.12": [-0.5967, -0.3873],
"-2.11": [-0.5894, -0.3805],
"-2.10": [-0.5816, -0.3743],
"-2.09": [-0.5734, -0.3685],
"-2.08": [-0.5648, -0.3633],
"-2.07": [-0.5560, -0.3587],
"-2.06": [-0.5468, -0.3547],
"-2.05": [-0.5374, -0.3513],
"-2.04": [-0.5278, -0.3484],
"-2.03": [-0.5181, -0.3462],
"-2.02": [-0.5082, -0.3447],
"-2.01": [-0.4982, -0.3437],
"-2.00": [-0.4883, -0.3434],
"-1.99": [-0.4783, -0.3437],
"-1.98": [-0.4683, -0.3447],
"-1.97": [-0.4584, -0.3462],
"-1.96": [-0.4487, -0.3484],
"-1.95": [-0.4391, -0.3511],
"-1.94": [-0.4296, -0.3545],
"-1.93": [-0.4204, -0.3584],
"-1.92": [-0.4115, -0.3629],
"-1.91": [-0.4028, -0.3678],
"-1.90": [-0.3945, -0.3733],
"-1.89": [-0.3865, -0.3793],
"-1.88": [-0.3788, -0.3858],
"-1.87": [-0.3716, -0.3927],
"-1.86": [-0.3648, -0.4000],
"-1.85": [-0.3584, -0.4077],
"-1.84": [-0.3524, -0.4157],
"-1.83": [-0.3470, -0.4241],
"-1.82": [-0.3420, -0.4328],
"-1.81": [-0.3376, -0.4418],
"-1.80": [-0.3336, -0.4509],
"-1.79": [-0.3302, -0.4603],
"-1.78": [-0.3273, -0.4699],
"-1.77": [-0.3250, -0.4796],
"-1.76": [-0.3232, -0.4895],
"-1.75": [-0.3219, -0.4994],
"-1.74": [-0.3212, -0.5094],
"-1.73": [-0.3211, -0.5194],
"-1.72": [-0.3215, -0.5293],
"-1.71": [-0.3224, -0.5393],
"-1.70": [-0.3238, -0.5492],
"-1.69": [-0.3258, -0.5590],
"-1.68": [-0.3283, -0.5687],
"-1.67": [-0.3313, -0.5782],
"-1.66": [-0.3348, -0.5876],
"-1.65": [-0.3388, -0.5968],
"-1.64": [-0.3433, -0.6057],

```

"-1.63": [-0.3482, -0.6144],
"-1.62": [-0.3535, -0.6229],
"-1.61": [-0.3593, -0.6310],
"-1.60": [-0.3655, -0.6389],
"-1.59": [-0.3720, -0.6464],
"-1.58": [-0.3790, -0.6536],
"-1.57": [-0.3862, -0.6605],
"-1.56": [-0.3939, -0.6670],
"-1.55": [-0.4018, -0.6731],
"-1.54": [-0.4100, -0.6788],
"-1.53": [-0.4185, -0.6841],
"-1.52": [-0.4272, -0.6890],
"-1.51": [-0.4361, -0.6935],
"-1.50": [-0.4453, -0.6975],
"-1.49": [-0.4546, -0.7011],
"-1.48": [-0.4641, -0.7043],
"-1.47": [-0.4737, -0.7070],
"-1.46": [-0.4834, -0.7093],
"-1.45": [-0.4933, -0.7111],
"-1.44": [-0.5032, -0.7125],
"-1.43": [-0.5131, -0.7134],
"-1.42": [-0.5231, -0.7139],
"-1.41": [-0.5331, -0.7139],
"-1.40": [-0.5431, -0.7135],
"-1.39": [-0.5531, -0.7127],
"-1.38": [-0.5630, -0.7114],
"-1.37": [-0.5728, -0.7097],
"-1.36": [-0.5826, -0.7076],
"-1.35": [-0.5923, -0.7050],
"-1.34": [-0.6018, -0.7021],
"-1.33": [-0.6112, -0.6987],
"-1.32": [-0.6205, -0.6950],
"-1.31": [-0.6296, -0.6908],
"-1.30": [-0.6386, -0.6863],
"-1.29": [-0.6473, -0.6815],
"-1.28": [-0.6558, -0.6763],
"-1.27": [-0.6641, -0.6707],
"-1.26": [-0.6722, -0.6648],
"-1.25": [-0.6801, -0.6587],
"-1.24": [-0.6877, -0.6522],
"-1.23": [-0.6950, -0.6454],
"-1.22": [-0.7021, -0.6383],
"-1.21": [-0.7089, -0.6310],
"-1.20": [-0.7154, -0.6234],
"-1.19": [-0.7217, -0.6156],
"-1.18": [-0.7276, -0.6075],
"-1.17": [-0.7332, -0.5993],
"-1.16": [-0.7385, -0.5908],
"-1.15": [-0.7436, -0.5821],
"-1.14": [-0.7482, -0.5733],
"-1.13": [-0.7526, -0.5643],
"-1.12": [-0.7567, -0.5552],
"-1.11": [-0.7604, -0.5459],
"-1.10": [-0.7638, -0.5365],
"-1.09": [-0.7669, -0.5270],
"-1.08": [-0.7696, -0.5174],
"-1.07": [-0.7721, -0.5077],
"-1.06": [-0.7741, -0.4979],
"-1.05": [-0.7759, -0.4880],
"-1.04": [-0.7774, -0.4782],
"-1.03": [-0.7785, -0.4682],
"-1.02": [-0.7793, -0.4582],
"-1.01": [-0.7797, -0.4483],
"-1.00": [-0.7799, -0.4383],
"-0.99": [-0.7797, -0.4283],
"-0.98": [-0.7793, -0.4183],
"-0.97": [-0.7785, -0.4083],
"-0.96": [-0.7774, -0.3984],
"-0.95": [-0.7760, -0.3885],
"-0.94": [-0.7744, -0.3786],

```

```
"-0.93": [-0.7724, -0.3688],  
"-0.92": [-0.7702, -0.3590],  
"-0.91": [-0.7676, -0.3494],  
"-0.90": [-0.7648, -0.3398],  
"-0.89": [-0.7617, -0.3303],  
"-0.88": [-0.7584, -0.3208],  
"-0.87": [-0.7548, -0.3115],  
"-0.86": [-0.7510, -0.3023],  
"-0.85": [-0.7469, -0.2932],  
"-0.84": [-0.7425, -0.2841],  
"-0.83": [-0.7379, -0.2753],  
"-0.82": [-0.7331, -0.2665],  
"-0.81": [-0.7281, -0.2579],  
"-0.80": [-0.7228, -0.2493],  
"-0.79": [-0.7174, -0.2410],  
"-0.78": [-0.7117, -0.2327]}}
```