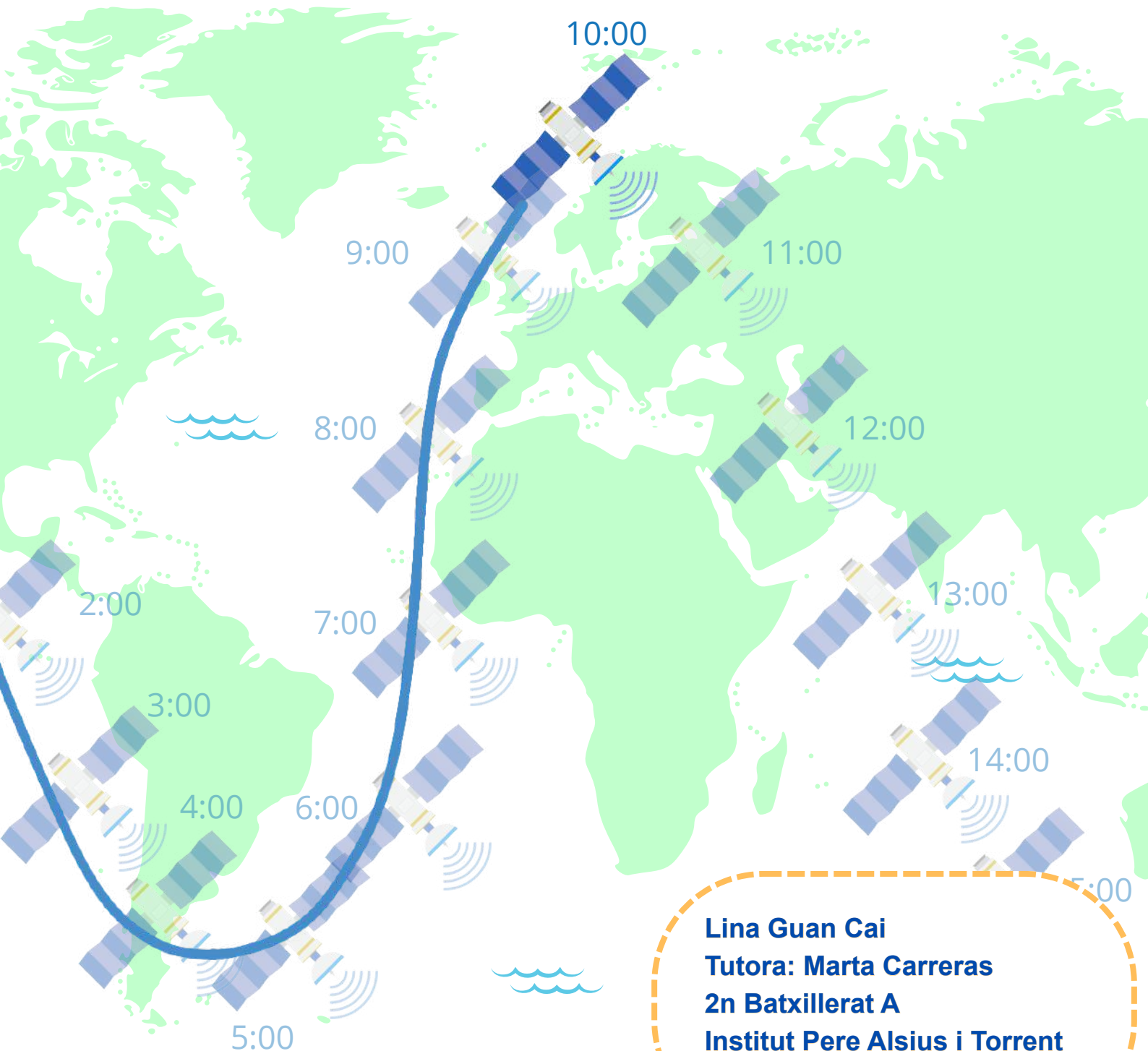


Treball de recerca

EXPLORANT EL CEL: PREDICCIÓ I SEGUIMENT DE SATÈL·LITS D'ÒRBITA BAIXA (LEO) MITJANÇANT UN SIMULADOR ORBITAL



Lina Guan Cai
Tutora: Marta Carreras
2n Batxillerat A
Institut Pere Alsius i Torrent
3 d'octubre de 2025, Banyoles

Agraïments

En primer lloc, vull agrair a la meva tutora del treball de recerca, Marta Carreras, per la seva orientació constant i els seus consells, que han estat essencials i em van ser de gran ajuda. També agraeixo al tutor del programa Argó, José A. López per proporcionar-me el codi del qual es va partir per dur a terme la part pràctica i per l'assessorament aportat.

Per últim, vull donar les gràcies especialment al suport de totes les persones que m'han estat al costat durant la realització d'aquest treball, a la família i als amics.

Abstract

Los satélites artificiales son inventos que han permitido una innovación tanto en los ámbitos científicos como en los tecnológicos. Hoy en día se han vuelto esenciales. El interés generado por ellos y las ganas de aprender ha dado paso a que el objetivo sea aplicar y profundizar en el uso del simulador orbital SGP4 para predecir y visualizar la trayectoria de algunos satélites de órbita baja. Se ha enfocado en los de este tipo ya que son los más utilizados y demandados en la actualidad por su cercanía a la Tierra. Para conseguirlo se ha partido del código SGP4 y se ha adaptado a lo que se necesitaba. Luego se ha diseñado un código para poder visualizar su posición en el futuro en un mapa 2D, creando como resultado una animación que muestra el recorrido que harán las naves durante un mes entero, desde el 25 de julio del 2025 al 24 de agosto del 2025. Desde el punto de vista de una persona que desea planear una observación aficionada para el 1 de agosto del 2025 se han analizado los resultados para dar respuesta a las hipótesis planteadas al principio, determinar los períodos de visibilidad para cada una de las naves y determinar el mejor momento para realizar la observación. Se ha comparado la predicción con los TLE que contienen la posición en la vida real de los satélites durante los días simulados para determinar la efectividad de la predicción. Se comentan las diferencias de trayectoria observadas entre diferentes satélites, los períodos de visibilidad, el momento que hay más satélites visibles y la efectividad de la predicción a lo largo de los 31 días.

Artificial satellites are inventions that have enabled innovation in both scientific and technological fields. Today, they have become essential. The interest they generate and the desire to learn has led to the objective of applying and deepening the use of the SGP4 orbital simulator to predict and visualize the trajectory of some low Earth orbit satellites. The focus has been placed on this type, since they are the most widely used and demanded today due to their proximity to Earth. To achieve this, the SGP4 code was taken as a starting point and adapted to the needs of the project. Then, a program was designed to visualize their future positions on a 2D map, resulting in an animation that shows the path the satellites will follow over an entire month, from July 25, 2025, to August 24, 2025. From the perspective of a person wishing to plan an amateur observation for August 1, 2025, the results were analyzed in order to address the hypotheses posed at the beginning, determine the visibility periods for each of the satellites, and identify the best moment to carry out the observation. Additionally, the prediction has been compared with the TLE that contains the real position of the satellites during the simulated days to determine the effectiveness of the prediction.

At the end of the project, the differences of trajectory observed between different satellites, the visibility periods and the moment when there are more satellites visible and the effectiveness of the prediction over the 31 days are discussed.

ÍNDEX

INTRODUCCIÓ	5
MOTIVACIÓ	5
OBJECTIUS	6
METODOLOGIA	7
MARC TEÒRIC	
1. INTRODUCCIÓ ALS SATÈL·LITS ARTIFICIALS	8
1.1 Història	8
1.1.1 Primers plantejaments	8
1.1.2 Primers satèl·lits	9
1.1.3. Primeres companyies	11
1.1.4 Fi de la cursa espacial	11
1.2 Introducció a la mecànica orbital	11
1.2.1 Lleis de Kepler	11
1.2.1.1 Primera Llei: Llei de les òrbites	12
1.2.1.2 Segona Llei: Llei de les àrees	13
1.2.1.3 Tercera Llei: Llei dels períodes	13
1.2.2 Elements orbitals	14
2. CLASSIFICACIÓ DELS SATÈL·LITS	15
2.1 Satèl·lits d'òrbita geoestacionària (GEO)	15
2.2 Satèl·lits d'òrbita terrestre mitjana (MEO)	16
2.3 Satèl·lits d'òrbita baixa (LEO)	16
3. TECNOLOGIA DELS SATÈL·LITS LEO	17
3.1 Sistemes bàsics	17
3.1.1 Estructura	17
3.1.2 Bateria	18
3.1.3 Orientació, navegació i control (GNC)	18
3.1.4 Comandament i gestió de dades	19
3.1.5 Control tèrmic	20
3.1.6 Escuts Whipple	20
3.1.7 Vela de desorbitació (drag sails)	20
3.2 Comunicacions	21

3.3 Disseny constel·lació satèl·lit.....	22
3.3.1 Constel·lació sense ISL (no inter-satellite link).....	23
3.3.2 Constel·lació amb ISL (inter-satellite link).....	23
3.4 Aplicacions.....	24
3.5 Exemples actuals.....	24
4. SISTEMES DE SEGUIMENT.....	26
4.1 Radar.....	26
4.2 Seguiment làser de satèl·lits (SLR).....	27
4.3 Fitxer TLE.....	28
4.3.1 Components.....	28
4.3.1.1 Número de línia.....	29
4.3.1.2 Número de catàleg.....	29
4.3.1.3 Classificació.....	29
4.3.1.4 Designador internacional.....	29
4.3.1.5 Època.....	30
4.3.1.6 Primera i segona derivada del moviment mitjà.....	30
4.3.1.7 Terme d'arrossegament BSTAR.....	30
4.3.1.8 Tipus efemèrides.....	30
4.3.1.9 Número d'element.....	30
4.3.1.10 Suma de verificació.....	30
4.3.1.11 Inclinació.....	31
4.3.1.12 Ascensió recta del node ascendent.....	31
4.3.1.13 Excentricitat.....	31
4.3.1.14 Argument Perigeu.....	31
4.3.1.15 Anomalia mitjana.....	31
4.3.1.16 Moviment mitjà.....	31
4.3.1.17 Número de revolució.....	31
<u>PART PRÀCTICA</u>	
5. SIMULADOR ORBITAL SGP4.....	32
5.1 Anàlisi del codi.....	32
5.2 Adaptació del simulador.....	37
5.2.1 Configuració de paràmetres.....	37

5.2.1.1 Punt de referència.....	37
5.2.1.2 Modificació del text imprès.....	39
5.2.1.3 Data.....	39
5.2.2 Satèl·lits per simular.....	40
6. CREACIÓ DE CODI PER VISUALITZAR LA TRAJECTÒRIA DELS SATÈL·LITS.....	41
6.1 Recerca d'informació.....	41
6.2 Descarregar llibreries.....	42
6.3 Primera versió.....	44
6.3.1 Importació de llibreries.....	44
6.3.2 Configuració del mapa 2D.....	44
6.3.3 Transformació de coordenades ECEF a latitud i longitud.....	45
6.3.4 Llista de latituds, longituds i creació de línia de la trajectòria.....	46
6.3.5 Lectura de dades i actualització del gràfic.....	47
6.3.6 Detecció de canvis bruscos.....	47
6.3.7 Pintat de la trajectòria.....	48
6.3.8 Avaluació.....	49
6.4 Segona versió.....	49
6.4.1 Importació de les llibreries i configuració del mapa.....	49
6.4.2 Funció transformació de coordenades ECEF a latitud i longitud.....	50
6.4.3 Processament de dades de posició.....	50
6.4.4 Línies de la trajectòria.....	51
6.4.5 Text.....	52
6.4.6 Funció “actualitzar”.....	52
6.4.7 Botons de control.....	55
6.4.7.1 Salts endavant.....	55
6.4.7.2 Pausar i avançar.....	56
6.4.7.3 Salt enrere.....	57
6.4.7.4 Connectar funcions als botons.....	58
6.4.8 FuncAnimation i mostrar dibuix.....	59
6.4.9 Comprovació.....	60
7. RESULTATS I INTERPRETACIÓ DE LES TRAJECTÒRIES DIBUIXADES DURANT EL DIA 1 D'AGOST.....	62

7.1 LandSat-9 (49260).....	62
7.2 LandSat-8 (39084).....	63
7.3 Terra (25994).....	64
7.4 Deimos-1 (35681) i Deimos-2 (40013).....	68
7.5 Starlink-1010 (44716).....	70
7.6 Oneweb-0012 (44057).....	71
7.7 Quantitat de satèl·lits visibles segons el temps a l'1 d'agost de 2025.....	72
8. EFECTIVITAT DELS RESULTATS OBTINGUTS AL LLARG DEL TEMPS.....	72
8.1 30 de juliol.....	73
8.2 4 d'agost.....	75
8.3 9 d'agost.....	77
8.4 14 d'agost.....	79
8.5 19 d'agost.....	82
8.6 24 d'agost.....	84
8.7 Recopilació de resultats i interpretació.....	86
9. CONCLUSIONS.....	88
10. FONTS CONSULTADES.....	92
10.1 Bibliografia.....	92
10.2 Webgrafia.....	92
11. ÍNDEX D'IMATGES.....	95
12. ANNEXOS.....	99
12.1 Annex 1: Simulador SGP4 proporcionat pel programa Argó de la Universitat Autònoma de Barcelona (fitxer comprimit).....	99
12.2 Annex 2: Simulador adaptat amb la 1a i 2a versió del codi per visualitzar la trajectòria (fitxer comprimit).....	99
12.3 Annex 3: Fitxers TLE dels satèl·lits simulats (PDF).....	99
12.4 Annex 4: main.py (fitxer principal) original (PDF).....	99
12.5 Annex 5: main.py (fitxer principal) adaptat (PDF).....	99
12.6 Annex 6: 1a versió del codi per visualitzar la trajectòria (PDF).....	100
12.7 Annex 7: 2a versió del codi per visualitzar la trajectòria (PDF).....	100
12.8 Annex 8: Gràfics.....	100
12.9 Annex 9: Mostra dels resultats (vídeo).....	100

INTRODUCCIÓ

MOTIVACIÓ

Els satèl·lits artificials són uns invents molt fascinants. Agafen inspiració en el moviment del nostre satèl·lit natural, la lluna, i a partir de les seves característiques físiques se n'han creat d'artificials perquè duguin a terme funcions que en anys anteriors mai es podrien haver imaginat. Gràcies a això s'ha pogut realitzar un gran avenç en el món tecnològic i científic. Ara aquests objectes poden arribar a tenir mides tan petites com una capsa de sabates o tan descomunals com 100 m de llargada i utilitats molt diferents s'han tornat essencials per la nostra societat.

Per començar, veient la quantitat que estan en òrbita i com cada vegada s'envien més satèl·lits a l'espai, podem arribar a la conclusió que en un futur dependrem encara més d'aquests invents i que la seva importància no deixarà d'augmentar. Sobretot els d'òrbita baixa, que són responsables de la major part de les aplicacions quotidianes. Per exemple, sense ells no existiria internet, ja que fan possible que totes les tecnologies estiguin connectades a una mateixa xarxa. Tampoc tindríem GPS, perquè aquest sistema és resultat de la seva capacitat de determinar la posició exacta d'una persona o no haguéssim obtingut tanta informació sobre l'univers.

Malgrat això, el que més m'atrau és la complexitat del seu funcionament i les seves òrbites, que són moviments que es poden veure influïts per molts factors. És per això que s'ha decidit centrar el treball en el seguiment i predicció dels satèl·lits, aprenent a predir trajectòries, identificar quins són visibles des de la nostra ubicació i comparar diferents constel·lacions fent ús d'una de les eines que fan servir els professionals per monitorar l'espai: el simulador orbital. És una eina que permet calcular la posició dels satèl·lits a partir de dades TLE (Two-Line Elements) durant el llarg del temps. Amb ella ens endinsarem dintre el món aeroespacial i representarem els resultats de manera gràfica, per tal de donar respostes a hipòtesis que foren fruit de la curiositat sobre com es desplacen aquests objectes i com poden ser vistos des de la Terra.

Ja des del principi se sabia que es volia enfocar el projecte en l'àmbit espacial perquè sempre ha estat un tema que m'ha captivat i que fins i tot m'he plantejat dedicar-m'hi en un futur. A partir d'aquí es va fer una recerca exhaustiva per determinar quin aspecte concret era més atractiu i el que em proporcionés més possibilitats per la part pràctica. En general, és un tema que em motiva molt i estic convençuda que serà una experiència enriquidora.

OBJECTIUS

Aquest treball girarà entorn d'aquest objectiu principal: Aplicar i aprofundir en l'ús del simulador orbital SGP4 per a predir i visualitzar la trajectòria de satèl·lits d'òrbita baixa.

Actualment, hi ha milers de satèl·lits d'òrbita baixa amb finalitats molt diverses: per a comunicacions, observació de la Terra, navegació i ciència que permeten el bon funcionament de la societat. Aleshores, conèixer la seva trajectòria i poder predir on es trobaran resulta essencial tant en l'àmbit tecnològic per evitar col·lisions contra altres naus i optimitzar el seu ús, com per l'àmbit científic per saber quan són visibles per poder realitzar les observacions, fotos i anàlisis necessàries. A més, també pot ser molt útil pels entusiastes que volen seguir els seus satèl·lits preferits i entendre millor el seu moviment.

Abans de res, s'ha reflexionat entorn el simulador orbital i s'han plantejat algunes hipòtesis que s'intentaran corroborar o refutar:

- És possible visualitzar les trajectòries dels satèl·lits en mapes 2D amb els quals serà possible veure on estaran a mesura que es vagi recurrent el temps durant 31 dies (1 mes).
- Totes les trajectòries LEO seran molt diferents les unes de les altres, les úniques que tindran la mateixa forma seran les que formen part de la mateixa constel·lació.
- És possible determinar els períodes de visibilitat per cada satèl·lit per un dia concret, la quantitat de minuts totals, el moment amb la quantitat màxima i l'òptim per dur a terme una observació aficionada.
- A mesura que passa el temps, els fitxers TLE utilitzats perdran precisió, s'aniran quedant obsolets i com a resultat el simulador orbital donarà resultats no exactes. No seran útils per fer prediccions acurades al cap de 5 dies.

Així doncs, per aconseguir aquest propòsit i donar resposta a les hipòtesis formulades es plantegen els següents objectius:

- Desenvolupar un programa partint del simulador orbital SGP4 que generi representacions visuals de les trajectòries dels diferents satèl·lits LEO escollits per estudiar en un mapa mundi 2D. Mostrar la seva posició i visibilitat en temps futurs. Determinar similituds i diferències en torn la forma de les seves rutes. Actualment, el programa no inclou la funció de visualitzar les naus de manera gràfica.
- Processar les dades de visibilitat del simulador des del punt de referència de l'institut Pere Alsius i Torrent per poder extreure conclusions. Identificar quines són les franges horàries on hi ha més satèl·lits LEO que estem predint visibles i la durada de

la visibilitat diària de cadascun de les naus per un dia concret. Definir el període òptim per realitzar una observació aficionada. El moment que es pugui veure a ull nu o bé amb telescopi el màxim de satèl·lits possible.

- Determinar la validesa d'aquest recurs. Comparar les posicions previstes obtingudes a partir de la propagació dels fitxers TLE del 25 de juliol amb els publicats en la data futura que hem fet la predicció per obtenir l'error comès i avaluar la seva gravetat.

METODOLOGIA

En primer lloc, per dur a terme un treball sobre l'estudi dels satèl·lits d'òrbita baixa s'ha hagut de buscar informació sobre la seva història, classificació i el seu funcionament a partir de llibres, documentals, pàgines web, publicacions i articles extrets de fonts fiables d'internet. Sempre s'ha comprovat que les pàgines web siguin d'institucions oficials o articles escrits per professionals de l'àmbit. S'ha començat introduint el concepte general de satèl·lit artificial i posat en context històric per tal d'obrir pas a explicacions més tècniques i enfocades als d'òrbita baixa.

En segon lloc, s'ha adaptat el simulador orbital SGP4 proporcionat per la Universitat Autònoma de Barcelona i s'ha escrit un propi codi per representar el moviment d'uns quants satèl·lits d'òrbita baixa en una animació amb Python. S'ha comprovat el seu bon funcionament. Llavors, s'han analitzat els resultats obtinguts i creat uns gràfics per donar resposta a les hipòtesis formulades a l'anterior apartat.

Finalment, s'ha avaluat l'efectivitat al llarg de 31 dies mitjançant la comparació entre les ubicacions previstes i la realitat. S'ha determinat la diferència presentada entre els dos valors i com ha anat evolucionant amb el temps.

MARC TEÒRIC

1. INTRODUCCIÓ ALS SATÈL·LITS ARTIFICIALS

Els satèl·lits artificials són màquines creades pels éssers humans que són llançades a l'espai mitjançant coets i que orbiten al voltant d'un cos celestial amb una finalitat pràctica. Actualment, n'hi ha més de 10.000 en òrbita al voltant de la Terra¹ i amb missions de tota mena. Això ens podria fer pensar que es van inventar fa molts anys o fins i tot a l'antiguitat, però la realitat és que no es van convertir en realitat fins a mitjans del segle XX.

1.1 Història

1.1.1 Primers plantejaments

El concepte de satèl·lit va sorgir per primer cop el segle XVI amb l'astrònom alemany Johannes Kepler que el va utilitzar per descriure les llunes que orbitaven al voltant de Júpiter. Va desenvolupar les tres lleis del moviment planetari i les seves taules astronòmiques precises van proporcionar proves del model heliocèntric de Copèrnic. Aquest model astronòmic el publicà el 1543 i situava el Sol al centre de l'Univers.

Una de les bases teòriques fonamentals que 200 anys més tard permetria concebre l'existència d'un satèl·lit artificial es remunta a l'any 1687, quan Isaac Newton va publicar l'estudi matemàtic *Philosophiæ Naturalis Principia Mathematica* que incloïa un experiment hipotètic, anomenat “bala de canó de Newton”, per explicar el moviment dels satèl·lits naturals. Imaginava un canó situat al pic d'una alta muntanya que dispara una bala, aquesta, seguiria una trajectòria diferent depenent de la velocitat inicial. Per exemple, si és baixa, el projectil pujarà, cada vegada més lentament, a causa de l'acceleració negativa de la gravetat i finalment caurà al lloc inicial on s'havia iniciat el moviment. Si és impulsada amb la rapidesa necessària per mantenir-se en òrbita a aquella altitud, recorreria una ruta circumfèrica al voltant de la terra i tornaria a la muntanya. En canvi, si es llancés a una velocitat més alta que l'orbital, però menor que la d'escapament, faria voltes al voltant de la Terra en una òrbita el·líptica. Aquesta analogia va ajudar a imaginar com funcionaria un satèl·lit i va establir un punt de partida per a futurs científics que van transformar les idees en càlculs pràctics.

¹ Dades extretes de: M. F. Melo. (2024, 10 octubre). Statista. *Hay más de 10.000 satélites activos sobre nuestras cabezas.*

<https://es.statista.com/grafico/33226/satelites-activos-en-orbita-alrededor-de-la-tierra/>

Segles més tard, al segle XIX, a causa d'enfrontaments polítics com la guerra freda i la II Guerra Mundial l'interès per l'exploració espacial va augmentar, ja que dominar-lo significava un avantatge estratègic en comparació amb la nació rival. La competència entre les dues potències de l'època, Rússia i Estats Units, fou coneguda com la cursa espacial.

Entre les investigacions realitzades per aconseguir un objecte volador va destacar el físic i matemàtic soviètic Konstatín Eduárdovich Tsiolkovsky amb l'equació del coet que va proporcionar una base matemàtica per calcular la velocitat necessària per posar un objecte en òrbita. Actualment, encara és utilitzada i s'ha convertit en un dels principis fonamentals de l'aeronàutica. A causa de les idees de gran importància aportades, se'l coneix com a pare d'aquest camp d'estudi.

Posteriorment, gràcies a la recerca prèvia, va poder sorgir el primer esbós detallat d'una estació espacial, l'autor del qual és en Herman Potočnik. A diferència d'en Konstatín que es va enfocar principalment en la teoria matemàtica dels coets, ell va posar més èmfasi a la infraestructura necessària per a una presència humana a l'espai. Va idear el disseny inicial mitjançant el càlcul de l'òrbita geoestacionària i, a més a més, va concebre un model inicial de comunicació entre diversos satèl·lits i la Terra, tot i que en aquell moment no li atorgà gaire importància.

1.1.2 Primers satèl·lits

El satèl·lit es va convertir en una realitat l'octubre del 1957. El primer fou Sputnik-1, d'òrbita baixa, enviat a l'espai per la Unió Soviètica. Dirigit per Sergei Korolev, un enginyer aeroespacial especialitzat en míssils balístics de gran abast, l'aparell era capaç de localitzar amb precisió els llocs de la superfície terrestre. Consistia en una esfera de plata amb un diàmetre de 56 cm i quatre llargues antenes. Tot i ser petit de mida, pesava 83 kg.



Fig. 1 *Sputnik-1*

Font: <https://www.latercera.com/culto/2024/10/07/sputnik-1-el-dia-en-que-se-lanzo-el-primer-satelite-al-exterior-o-como-se-desato-la-carrera-espacial/>

L'èxit del llançament va suposar un avenç tecnològic significatiu pel bàndol soviètic. Prèviament, treballaven en un de més gran, Object-D, des de 1956, però en veure la lentitud que presentava el desenvolupament van optar per una alternativa més simple que poguessin enviar de manera ràpida i que funcionés durant un període curt fins que aconseguissin crear un satèl·lit més complex. Com a resultat, Sputnik-1 només va romandre en funcionament durant 22 dies, aleshores es van esgotar les piles, va caure a l'atmosfera i es va destruir. La por que els nord-americans aconseguissin aquest triomf primer també els va impulsar a prendre aquesta decisió.

Un mes més tard, abans que els estatunidencs poguessin enviar el seu primer satèl·lit, els soviètics van tornar a guanyar avantatge sobre ells amb l'Sputnik-2. A l'interior hi transportava una gossa anomenada Laika. Malauradament, no van aconseguir desenvolupar una estratègia de reentrada abans de la missió i van planificar deixar-la viva fins que s'esgotés el seu sistema de suport vital. No va arribar lluny, ja que va morir en poques hores a causa de les condicions extremes del vol. Tant l'Sputnik-2 com la Laika es van desintegrar a l'atmosfera terrestre l'abril de 1958.

Amb els dos èxits del bàndol Soviètic, els nord-americans tenien la pressió d'atrapar-los a nivell de desenvolupament tecnològic i posar-se al dia. Estaven treballant en el seu propi projecte, el Vanguard, i addicionalment també n'havien aprovat un altre anomenat Explorer.

Aquest primer, que inicialment pretenia ser un sistema avançat amb un coet de tres etapes basat en noves tecnologies, es va veure afectat per nombrosos retards i fracassos tècnics. Tenien previst llançar-lo a finals de 1957 i que provés les capacitats de llançament del Vanguard, així com estudiar els efectes de l'entorn espacial sobre el satèl·lit, però va fracassar quan el coet que el transportava explotà pocs segons i el feu malbé. Amb posterioritat a aquest desastre, es van enfocar més en el programa Explorer i van poder aconseguir el seu primer èxit el 31 de gener de 1958. Fou el primer a transportar instruments científics. Portava un detector de rajos còsmics, dissenyat per James Van Allen, per mesurar la radiació de l'entorn de l'atmosfera de la Terra. Mentre era en òrbita, va detectar una quantitat de raigs còsmics més baixa del que s'esperava. Més tard, es va demostrar que es devia a l'existència d'un cinturó de radiació proper a l'atmosfera terrestre.

D'altra banda, el projecte Vanguard no va veure la llum fins al 17 de març de 1958 quan finalment el van aconseguir enviar. Tot i no ser el pioner com havien volgut, fou el primer a utilitzar cèl·lules solars per a l'alimentació. Encara avui continua en òrbita i és el satèl·lit artificial més antic que roman a l'espai.

1.1.3. Primeres companyies

L'any 1962 es va fundar la COMSAT Corporation als Estats Units, la primera companyia dedicada exclusivament a aquest àmbit. Va marcar un punt important en el desenvolupament de les telecomunicacions per satèl·lit. Aquest mateix any es van llançar els satèl·lits Telstar-1 i posteriorment, Telstar-2 que van permetre realitzar les primeres comunicacions transatlàntiques experimentals. Aquestes transmissions es van dur a terme entre diverses estacions terrestres situades a Goonhilly Down (Regne Unit), Pleumeur-Bodou (França), Andover i Maine (Estats Units).

Els avenços mencionats anteriorment no només van fer possible la comunicació de veu entre diferents bases, sinó que també van obrir la porta a la transmissió d'imatges en directe via satèl·lit, un fet revolucionari per a l'època. A partir d'aquest punt, nombrosos satèl·lits van ser dissenyats per descobrir les limitacions d'actuació dels sistemes de comunicació.

Europa es va iniciar al món aeroespacial l'any 1975, el mateix any en què es va crear l'estació espacial europea amb el llançament de Cos-B. Fou un satèl·lit amb la missió de cartografiar el cel, en particular la Via Làctia, i estudiar els orígens dels rajos gamma amb una energia superior als 50 MeV. Dissenyat amb una resistència de dos anys, va sorprendre el món quan es va mantenir en òrbita durant sis anys, quatre més dels esperats.

1.1.4 Fi de la cursa espacial

El 1975 també es va acabar oficialment la cursa espacial. Els Estats Units i l'URSS van dur a terme la primera missió espacial internacional amb l'objectiu de demostrar pau entre les dues superpotències mundials. Va consistir a unir dues naus a l'espai (Apollo-Soyuz). Marcà l'inici de la cooperació internacional i va obrir camí a projectes internacionals.

Un d'aquests projectes més importants fou l'ISS (International Space Station) que va sorgir gràcies a la cooperació de les dues potències mundials de l'època, EUA i Rússia, amb els següents països i continents: Japó, Canadà i l'EU que actualment s'utilitza com a laboratori de recerca de la microgravetat i ambient espacial.

1.2 Introducció a la mecànica orbital

1.2.1 Lleis de Kepler

Són un conjunt de tres teoremes que descriuen el moviment dels planetes i altres objectes que orbiten al nostre sistema solar. S'utilitzen per explicar les formes de les trajectòries

orbitals, comprendre els canvis de velocitat dels planetes en moviment i calcular propietats com el període orbital i la distància mitjana entre un planeta i una estrella.

1.2.1.1 Primera llei: llei de les òrbites

Acaba amb la idea que defensava Copèrnic, que les òrbites havien de ser perfectament circulars. Estableix que tots els planetes giren seguint una trajectòria el·líptica, que es va repetint al voltant del Sol i aquest darrer no se situa al centre sinó en un dels focus de manera fixa. De la mateixa manera que defineix les característiques d'una el·lipse, una figura geomètrica semblant a un cercle allargat.

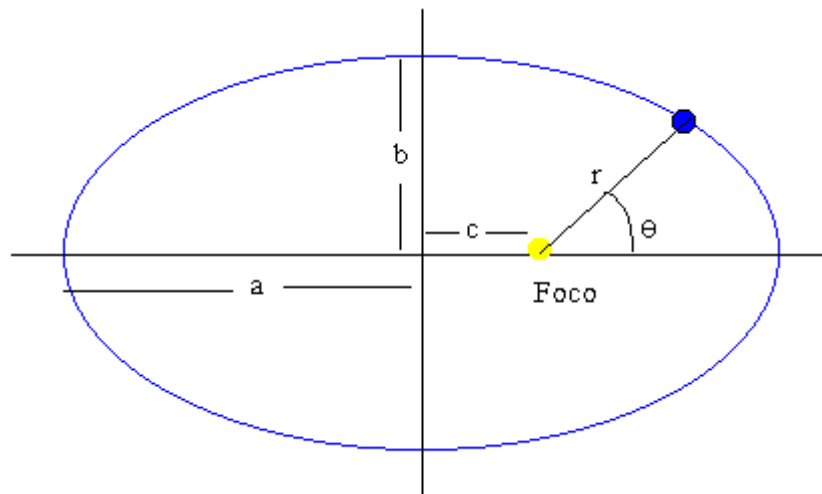


Fig. 2 Parts d'una el·lipse

Font: <http://www.sc.ehu.es/sbweb/fisica/celeste/kepler/kepler.htm>

Consta de dos focus: dos punts fixos dins de l'el·lipse. En el cas de les òrbites planetàries, el Sol n'ocupa un.

r_1 (periheli) és la distància més propera al focus, en altres paraules quan l'angle θ és 0, és el punt de l'òrbita on el planeta està més a prop del Sol.

r_2 (afeli) és la distància més allunyada del focus, quan l'angle θ és igual a π , és el punt de l'òrbita on el planeta està més lluny del Sol.

a correspon al semieix major, la meitat de l'eix més llarg d'una el·lipse que pot ser calculat amb la següent equació:

$$a = (r_1 + r_2)/2$$

La suma del periheli amb l'afeli correspondria a l'eix major, a la longitud total del diàmetre llarg de l'el·lipse, quan ho dividim entre 2 trobem la meitat d'aquest eix.

b correspon al semieix menor, la meitat de l'eix més curt d'una el·lipse.

c correspon a la semidistància focal, que va des del centre de la figura fins al focus i ve donada per la fórmula:

$$c = (r_2 - r_1)/2$$

Restem el periheli a l'afeli i ho dividim entre dos per saber la meitat de la diferència entre les dues distàncies.

La relació dels semieixos seria:

$$a^2 = b^2 + c^2$$

L'excentricitat representada amb la lletra ϵ mesura com n'és d'allargada la figura. Es calcula de la manera següent:

$$\epsilon = \frac{c}{a} = \frac{(r_2 - r_1)}{(r_2 + r_1)}$$

Dividim la semidistància focal (c) entre el semieix major (a). Només serà una el·lipse si l'excentricitat té un valor entre 0 i 1. Si la c és igual 0, tindrà una excentricitat de 0. Voldrà dir que els focus estaran al centre, per tant, serà un cercle perfecte. Si és major a 1 tindrem una hipèrbola, es tractaria d'una òrbita oberta on l'objecte no torna mai. I finalment si és igual a 1, tindríem una paràbola.

1.2.1.2 Segona Llei: Llei de les àrees

Postula que el moment angular (L), el producte de la massa del planeta, la seva velocitat i la distància al centre del sol, es conserva. Com que L sempre ha de ser el mateix, si el planeta està més lluny, la distància (r_1) és major, la seva velocitat (v_1) disminuirà, mentre que si està més a prop, la distància (r_2) és menor, la seva velocitat (v_2) augmentarà. Resumint, les dues magnituds són inversament proporcionals, el planeta accelera quan s'apropa al Sol i frena quan s'allunya. L'enunciat s'expressa amb la igualtat següent:

$$L = m * r_1 * v_1 = m * r_2 * v_2$$

1.2.1.3 Tercera Llei: Llei dels períodes

Els quadrats dels períodes de revolució (T) són directament proporcionals als cubs dels semieixos majors (a) de l'el·lipse. La k és la constant de proporcionalitat que depèn del sistema de referència i unitats utilitzades.

$$T^2 = k * a^3$$

1.2.2 Elements orbitals

Els elements orbitals són un conjunt de sis paràmetres que es necessiten per identificar i descriure de manera única una òrbita concreta. Són molt útils en astronomia i mecànica celeste perquè permet saber no només com és l'òrbita, sinó també la localització de l'objecte i el seu desplaçament al llarg del temps. Generalment, es consideren en sistemes clàssics de dos cossos, on s'utilitza una òrbita kepleriana.

Ara bé, quan observem el moviment des d'un sistema inercial, fix, veurem que cada cos segueix una trajectòria diferent i cadascuna d'aquestes té el seu focus en el centre de massa comú. Tanmateix, quan es veu d'un sistema de referència no inercial centrat en un dels cossos, només es veu la trajectòria del cos oposat. Donat un marc de referència inercial i una època arbitrària (un punt especificat en el temps) calen sis paràmetres per definir sense ambigüitats una òrbita arbitrària i impertorbable. Això és degut al fet que el problema conté sis graus de llibertat que corresponen a les tres dimensions espacials que defineixen la posició, xyz en un sistema de coordenades cartesianes, més la velocitat en cadascuna d'aquestes dimensions. Aquestes es poden descriure com a vectors d'estat orbital, però sovint és una manera inconvenient de representar una òrbita, motiu pel qual s'utilitzen els elements orbitals keplerians. Consisteixen en els següents:

- Època: moment exacte en què s'han mesurat les dades de l'òrbita. Les òrbites poden variar de manera lleugera amb el temps, per això és important saber el moment en què es va registrar. Seria com posar la data i hora.
- Excentricitat: allargament de l'òrbita (vegeu apartat 1.2.1.1)
- Semieix major: indica grandària de l'òrbita de manera general (vegeu apartat 1.2.1.1)
- Inclínació (i): Inclínació vertical de l'el·lipse respecte al pla de referència expressada en graus. Si $i=0^\circ$ serà completament plana.
- Ascensió recta del node ascendent: angle que ens diu on es troba el punt on el satèl·lit passa de sota a sobre del pla de referència. Aporta informació sobre la direcció horitzontal de l'òrbita a l'espai.
- Argument de perigeu (w): Orientació de l'el·lipse en el pla orbital, com a angle mesurat des del node ascendent fins al perigeu.
- Anomalia veritable (v, θ o f): defineix posició del cos en òrbita al llarg de l'el·lipse en un moment específic.

2. CLASSIFICACIÓ DELS SATÈL·LITS

Actualment, hi ha molts tipus de satèl·lits, de manera que s'han ideat diverses formes de classificar-los. La més comuna de totes és segons la seva altitud orbital, la distància de la superfície terrestre que afecta la velocitat en què es desplacen i l'àrea que incideixen. Aleshores, a l'hora d'escollir el tipus d'òrbita, s'hauran de tenir en compte la finalitat i les funcions desitjades.

2.1 Satèl·lits d'òrbita geoestacionària (GEO)

Els satèl·lits GEO (*Geostationary Earth Orbit*) són els més comuns avui en dia situats a una altitud orbital d'uns 36000 km, just per damunt de l'equador. El seu període orbital coincideix amb el període de rotació de la Terra, 23 hores, 56 minuts i 4 segons, fet que els permet sempre mantenir una posició fixa respecte a un determinat punt del planeta. Quan s'observen des de la superfície terrestre sembla que estiguin immòbils. Això també permet que no siguin necessàries les transferències de connexions que en el cas dels altres tipus de satèl·lit sí que ho són. Aquesta característica fa que siguin avantatjosos a l'hora de proporcionar serveis de cobertura contínua. Per exemple, s'utilitzen en telecomunicacions i transmissions de televisió, ja que cada un d'aquest tipus d'òrbita és capaç de cobrir una gran superfície. Així doncs, tres satèl·lits serien els que es necessitarien per abastar la superfície de la Terra.

No obstant això, a causa de les seves majors altituds orbitals, requereixen més energia de llançament i sistemes més complexos per mantenir-los. A part d'això, presenten més latència i una velocitat de transmissió menor que els satèl·lits MEO i LEO.

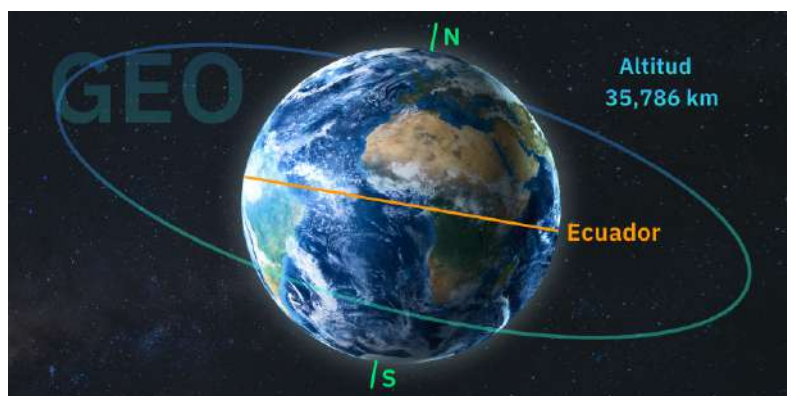


Fig. 3 Esfera terrestre amb les coordenades equatorials, altitud geoestacionària
Font: <https://eos.com/blog/types-of-satellites/>

2.2 Satèl·lits d'òrbita terrestre mitjana (MEO)

Els satèl·lits MEO (*Medium Earth Orbit*) poden ser trobats de 5000 km a 20000 km d'altitud sobre la superfície terrestre. El seu període orbital és d'entre 2 i 12 hores. Són utilitzats especialment per serveis de navegació i posició com ara GPS, ja que ofereixen un punt intermedi entre l'àrea de cobertura i la velocitat de transmissió de dades. Per un costat, mentre que els de baixa òrbita necessitarien milions de satèl·lits per aconseguir una cobertura total de la Terra, amb els d'aquest tipus només se'n requeririen 6. La transferència de connexió continuaria sent necessària, però el nombre de vegades que caldrà serà menor que els GEO en tenir un rang de visió major. Però, per un altre costat, els senyals serien més febles i presentarien un retard més alt en el temps que tarden les dades a arribar al satèl·lit i tornar (latència).

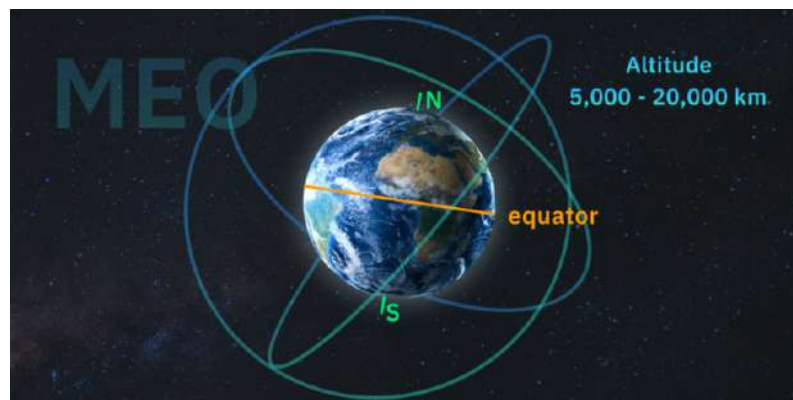


Fig. 4 Esfera terrestre amb les coordenades equatorials, altitud mitjana
Font: <https://eos.com/blog/types-of-satellites/>

2.3 Satèl·lits d'òrbita baixa (LEO)

Els satèl·lits LEO (*Low Earth Orbit*) es mouen a una altitud de 160-1500 km per sobre de la superfície terrestre, que correspon a la zona de la termosfera i la part més baixa de l'exosfera. Sota aquests nivells, l'òrbita decauria ràpidament a causa de la resistència de l'atmosfera terrestre, fet que restringeix les que són molt baixes a missions balístiques de curt termini o trajectòries propulsades. És el tipus que vola més proper a la superfície terrestre, degut això ofereix una cobertura inferior a altres menes de satèl·lit, però alhora fa que posseeixin una latència menor. Grups de satèl·lits, anomenats constel·lacions, són enviats de manera conjunta per formar una xarxa al voltant del planeta i així col·laborar per formar cobrir àrees més extenses. En ser no estacionaris, tenen la llibertat de moure's cap al lloc on són necessaris. Els períodes orbitals són curts, és capaç de fer la volta a la Terra fins a 16 vegades al dia. Són els més adequats per a tota classe de teledetecció, investigació científica o observació, ja que permet que les dades puguin ser transmeses i adquirides

ràpidament. Generalment, els costos de producció d'aquest tipus de naus són menors perquè acostumen a ser més petits. Una gran part són nanosatèl·lits que tenen massa i mides reduïdes. No acostumen a superar els 10 kg.

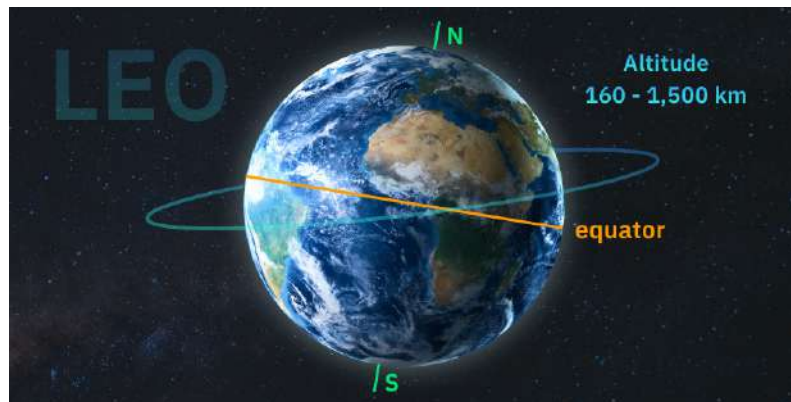


Fig. 5 Esfera terrestre amb les coordenades equatorials, altitud baixa
Font: <https://eos.com/blog/types-of-satellites/>

3. TECNOLOGIA DELS SATÈL·LITS LEO

3.1 Subsistemes bàsics

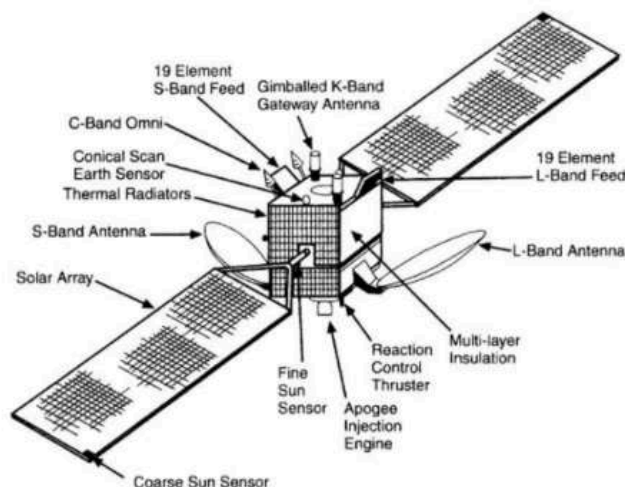


Fig. 6 Esquema d'un satèl·lit
Font: <https://images.app.goo.gl/eLP9kmAJmWS7VnH39>

3.1.1 Estructura

L'estructura de la nau ha de proporcionar una plataforma estable i forta per poder carregar instruments i subsistemes. Gràcies a la seva rigidesa és capaç de suportar grans tensions mecàniques imposades durant les diferents fases de les missions. Tota l'estructura es fa de materials molt lleugers com ara l'alumini combinat amb fibres de vidre. Actualment, aquests darrers són els més utilitzats a causa de la seva lleugeresa i resistència. El titani i altres

polímers reforçats amb fibra de carboni que són matèries que disposen d'una expansió tèrmica baixa també poden ser emprats. Si es compara l'estructura de la majoria de satèl·lits LEO amb els d'òrbita geoestacionària és relativament més petita. Normalment, no superen els 1000 kg, tot i que hi ha algunes excepcions com els de missions militars. Dels que són nanosatèl·lits destaca el CubeSat. Es caracteritza per tenir l'estructura en forma de cub, característica que li permet ser integrat en un llançador que tindrà la responsabilitat de deixar-lo anar a l'òrbita corresponent a l'espai. El més petit pot arribar a tenir la mida d'un cub de Rubik (10 cm x 10 cm x 10 cm) i pesar 1,3 kg.

3.1.2 Bateria

Els satèl·lits LEO s'alimenten d'energia solar captada a través de panells solars (*solar arrays* en fig.6) que es despleguen un cop en òrbita mitjançant cardans (*gimbals*): suports giratoris i mecanismes de retenció i alliberament responsables d'obrir-los a l'espai i mantenir-los segurs durant el llançament.

A l'espai s'orienten de manera perpendicular als rajos solars obtenint la màxima energia possible. Aquesta energia a més d'utilitzar-se a l'instant per alimentar tots els mecanismes de nau, també s'emmagatzema en bateries que s'autocarreguen i és aprofitada durant els períodes d'eclipsi (quan el sol està ocult) o quan hi ha una demanda baixa. Sistemes de regulació, commutació i distribució són les encarregades de subministrar electricitat a tots els sistemes electrònics.

3.1.3 Orientació, navegació i control (GNC)

Les tres funcions principals del sistema són decidir quin camí ha de seguir un satèl·lit (guia), calcular on és, a quina velocitat es mou (navegació) i realitzar els ajustos perquè segueixi la ruta i s'orienti segons les necessitats de la missió (control). Està format per tres components:

- Control d'actitud i òrbita (AOCS): encarregat de controlar l'orientació i l'òrbita del satèl·lit. S'utilitzen sensors: solars (*Fine Sun Sensor* i *Coarse Sun Sensor* en la fig. 6), d'escaneig cònic de la Terra (*Conical Scan Earth Sensor* en la fig. 6), receptors que reben i processen els senyals GPS que arriben per l'antena L-band (*L-band antenna* i *19-element L-band feed* en la fig. 6) i unitats de mesura inercial, totes proveeixen dades de la nau respecte a diferents cossos celestials per poder obtenir informació més precisa sobre el seu posicionament i el moviment. D'actuadors s'usen els següents: magnetòmetres que proporcionen una actitud aproximada i un model del camp magnètic terrestre, i rodes de reacció que generen torques

giroscòpics, rotacions oposades al centre de massa, fet que permet un canvi de direcció.

- Subsistema de propulsió: estableix i corregeix errors en la trajectòria amb petits propulsors (*Reaction Control Thruster* en la fig. 6) i amb el motor d'injecció d'apogeu (*Apogee Injection Engine* en la fig. 6) s'encarrega de col·locar el satèl·lit a l'òrbita desitjada després del llançament.
- Algoritmes integrats: processen les dades dels sensors i determina les accions de control que cal aplicar en cada moment.

3.1.4 Comandament i gestió de dades

Atorguen la capacitat d'enviar dades de manera contínua a la Terra (enllaç descendent) i fer un seguiment de la salut de la nau per assabentar-se de qualsevol avaria mitjançant un sistema que inclou funcions de telemetria, monitoratge, comandament i control. Així mateix, com faciliten la recepció de comandaments des del planeta (enllaç ascendent) per controlar les funcions i assignacions de la nau, envien instruccions per executar tasques com activar o desactivar transceptors, commutar entre unitats redundants, activar pirotècnics com separació de fases o desplegaments mecànics. Les antenes són els components principals d'aquest sistema. Poden ser de diferents tipus segons la seva freqüència:

Tipus	Freqüència	Ús
L-band	1-2 GHz	GPS, navegació, dades de baixa velocitat, telèfons satel·litals
S-band	2-4 GHz	Comandament, telemetria, comunicació de dades en satèl·lits LEO. (<i>S-band antenna</i> en fig.6)
C-band	4-8 GHz	Comunicacions per satèl·lit, transmissió TV, ràdio, més resistent a la pluja (<i>C-Band Omni</i> en fig.6)
X-band	8-12 GHz	Comunicacions militars, radar, dades científiques d'alta velocitat.
Ku-band	12-18 GHz	Comunicacions per satèl·lit comercials, TV per satèl·lit, internet satel·lital.
K-band	18-27 GHz	Comunicacions d'alta velocitat, experiments científics

		(<i>Gimballed K-band gateway antenna</i> en fig.6).
Ka-band	26-40 GHz	Comunicació per satèl·lit d'alta capacitat, radars d'alta resolució i curta distància, avions militars.

A vegades poden venir acompanyats d'un conjunt d'alimentació (*feed array*), un sistema complex d'antenes d'alimentació que permet controlar amb precisió la fase i amplitud del senyal que arriba a cada element del dispositiu de recepció, la qual cosa condueix a una alta directivitat i, per tant, amb pèrdues mínimes. Tanmateix, la fabricació pot ser costosa i, per això, no s'apliquen a totes, només s'utilitzen en sistemes de radar i en els de comunicació satel·lital.

3.1.5 Control tèrmic

L'efecte de temperatures altes a conseqüència del Sol i de les baixes durant els eclipsis poden causar tensions que podrien provocar fractures o deformacions plàstiques en els materials. El control tèrmic permet aconseguir un balanç adequat i garantir un bon desenvolupament de subsistemes mitjançant l'ús de mantes aïllants (*Multi-layer Insulation (MLI)* en fig.6) que minimitzen la pèrdua de calor i protegeixen contra condicions extremes, tubs que transfereixen l'escalfor lluny de zones sensibles, radiadors (*Thermal Radiator* en fig.6) encarregats de dissipar el seu excés cap a l'espai i dispositius de control tèrmic format per sensors i circuits d'escalfament. Tots aquests components permeten la protecció dels equips del bus del satèl·lit i una bona regulació de temperatura per tal d'aconseguir un rendiment eficient.

3.1.6 Escuts Whipple

En aquests últims anys, per culpa de la gran quantitat de naus que s'han llançat a l'espai, la quantitat de deixalles especials (MMOD) no ha deixat d'augmentar. Cada vegada està esdevenint més congestionat. Actualment, s'empren escuts Whipple en satèl·lits d'òrbita baixa, una placa d'alumini amb una distància de separació establert com a protecció per reduir l'impacte operatiu de les col·lisions amb MMOD, perquè és la zona més contaminada on el risc és molt alt. Es fabriquen i s'instal·len mentre són a la Terra. En GEO i MEO hi ha menys residus pel que de moment no necessiten la protecció.

3.1.7 Vela de desorbitació (*drag sails*)

La reducció de MMOD és un aspecte a tenir en compte després de l'observació de les seves greus conseqüències. La solució que han trobat per evitar que els satèl·lits abandonats es

quedin a l'espai per sempre és incorporar una vela de desorbitació que augmenta la superfície frontal exposada. Això facilita una sortida d'òrbita ràpida aprofitant la fricció atmosfèrica com a força principal. Cal destacar que és un sistema que només és adequat per satèl·lits d'òrbita baixa (LEO) fins a una massa d'una tona. Per la minoria de casos especials que pesen més, es fan servir altres tecnologies. Finalment, no s'utilitzen aquests sistemes per a tipus d'òrbites superiors, ja que la densitat és massa baixa perquè la fricció sigui efectiva per desviar-los.

3.2 Comunicacions

Un típic sistema de comunicació seria com el que apareix a la imatge.

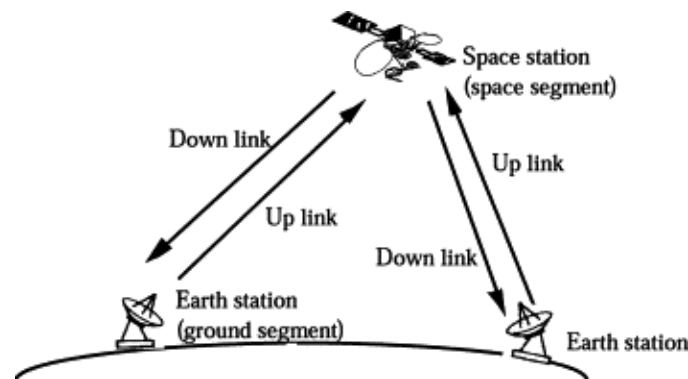


Fig. 7 Sistema de comunicació típic

Font: <https://www.sciencedirect.com/topics/engineering/satellite-communication-system>

La xarxa terrestre correspon a senyals de veu, dades o vídeos provinents de la telefònica pública commutada o d'una de telefonia mòbil. Conseqüentment, una estació a terra (també coneguda com a *gateway* en la tecnologia mòbil) és emprada per processar senyals abans que siguin transmesos al satèl·lit mitjançant un protocol d'enllaç ascendent (*uplink*).

A través de la xarxa terrestre es reben transmissions com serien canals de veu, canals analògics i altres dades digitals que combinats formen senyals de base. Són la informació original que es vol comunicar, però que encara no han estat processats de tal manera per poder ser enviats cap al satèl·lit. Si es continuen combinant es crea un únic senyal de base amb alta complexitat que inclou tot l'anterior. La barreja dels de base simple amb d'altres d'orígens diferents és conegut com a multiplexació. Es tracta d'un procés que estalvia espai i permet enviar més informació a la vegada.

Paral·lelament, es genera una ona sinusoidal mitjançant un oscil·lador situat també a la superfície terrestre amb la funció de transportador. Posteriorment, és modulada, que consisteix en la modificació de la seva amplitud, la fase o la freqüència, perquè pugui portar la informació empaquetada en els senyals de base complexa cap al satèl·lit.

Aquest procés pot ser de dos tipus:

- Analògic: si presenta continuïtat, que pot tenir valors infinits dins un interval que permet que pugui imitar la forma natural de les dades. Això fa que sigui molt precisa en la transmissió.
- Digital: si és discreta, no presenta continuïtat a causa del fet de només poder tenir dos valors. Normalment, els dos que corresponen al codi binari: 0 i 1. En comptes d'imitar la forma natural de la informació, si és digital es codifica en bits. Això es tradueix en el fet que és menys precisa i hi ha la possibilitat de perdre detalls. Tot i això, aquest tendeix a ser més comú que l'altre gràcies a la seva major capacitat de correcció d'error i insensibilitat al soroll.

Així doncs, el resultat final és un únic senyal compost per un de banda complexa que conté totes les dades per a ser enviades incrustades en una de sinusoidal que actua com a portadora d'informació. Com que ha de recórrer una distància considerable, el que es vol comunicar ha de passar per un procés d'amplificació on s'incrementa la seva amplitud per assegurar que arribi al satèl·lit amb suficient potència, ja que durant el trajecte pot experimentar atenuació (pèrdua de potència) a causa d'interferències, dispersió, absorció atmosfèrica, fet que podria causar que es perdessin dades.

Després és enviada al satèl·lit que el rep físicament amb l'antena. El transponedor amb el qual està connectat el filtra per eliminar interferències, el torna a amplificar de nou perquè tingui prou potència per ser retransmès cap a la Terra, modifica la freqüència per evitar interferències amb l'enllaç ascendent i finalment l'envia cap a l'estació terrestre receptora.

Quan és captada per l'antena de la terminal és molt feble, per tant, cal amplificar-lo. Això es duu a terme mitjançant un amplificador de baix soroll (LNA) que l'enforteix sense afegir cap mena de distorsió assegurant el manteniment de qualitat.

Un cop és prou fort, es desmodula i desmultiplexa, que són l'invers dels primers processos pels quals han estat sotmesos per poder fer transmissions amb l'enllaç ascendent. El primer modifica la seva amplitud i la freqüència a l'original, per tal de recuperar la informació i el segon separa els diferents tipus de dades que han estat combinats.

3.3 Disseny constel·lació satèl·lit

A l'hora de dissenyar la constel·lació LEO que consisteix en nombrosos satèl·lits en òrbita de 500-2000 km es té en compte tres factors principals:

- Cobertura global

- Cost d'un sol satèl·lit i la construcció de tota la constel·lació
- Escenari d'aplicació que es té com a objectiu

Segons les característiques anteriors la constel·lació es pot classificar en dues classes:

3.3.1 Constel·lació sense ISL (no inter-satellite link)

Els satèl·lits no presenten enllaços entre ells, no es poden passar informació de manera directa, així que només poden enviar dades quan sobrevolen una estació terrestre.

Aquest tipus de constel·lació és adequada per aplicacions amb retard tolerable (DTA), cost i complexitat més baixos com en el cas de l'observació terrestre i la previsió meteorològica on no és necessari transmetre la informació en temps real.

Durant el disseny s'ha de tenir en compte l'excentricitat de l'òrbita, la inclinació i l'altitud, ja que tindran una influència en el temps i l'àrea de cobertura del satèl·lit.

En el cas de ser el·líptiques, quan estan a prop de l'apogeu, el punt més lluny del planeta, l'àrea i el temps augmentaran de manera relativa. Cobrirà un espai major perquè se situarà a una altitud més gran, conseqüentment també hi passarà més estona, ja que segons la llei de Kepler es mou més lentament. Tanmateix, quan estan a prop del perigeu, el punt més a prop de la Terra, succeirà el contrari, els dos valors anteriors disminuiran alhora. La zona que cobrirà serà menor, per culpa de la seva menor altitud i hi passarà menys estona, ja que es mou de manera més ràpida en aquesta part.

Aquestes irregularitats poden complicar molt la planificació de les estacions terrestres, i per això normalment es prefereixen les òrbites circulars, ja que aquestes garanteixen que els satèl·lits tinguin una mateixa potència i passin pel mateix lloc en una mateixa hora, faciliten la connexió i la potència necessària per comunicar. Així doncs, també es dissenyen perquè siguin repetitives, és a dir, que passin de nou pel mateix lloc després d'un cert nombre de dies, per facilitar el control durant el funcionament. Es podrà saber on seran en certs punts i preparar la comunicació.

3.3.2 Constel·lació amb ISL (inter-satellite link)

Per implementar els enllaços intersatèl·lits (ISLs) i aconseguir una cobertura global sense interrupcions, la constel·lació LEO tendeix a orbitar en trajectòries que passen pels pols (òrbita polar) de manera que van de nord a sud en cada volta i aconsegueixen cobrir tot el planeta. No obstant això, pot generar un problema, si totes passen pel mateix punt als pols, amb el mateix angle d'inclinació estàndard de 90° , podria donar lloc a xocs. Per resoldre

aquest conflicte es modifica l'angle, en comptes de fer que totes siguin de 90°, se'ls assigna una inclinació lleugerament diferent, que sol anar dels 80° als 100° (excepte els 90°). Com a resultat, els satèl·lits no passaran pel mateix punt i estaran separats. Evitaran així la possibilitat de xoc. Mentrestant, per reduir el cost de construcció, es busca que la constel·lació tingui el mínim nombre de satèl·lits necessaris per aconseguir la cobertura global.

3.4 Aplicacions

Els satèl·lits LEO són emprats en àmbits molt diversos gràcies a la seva versatilitat i la seva capacitat d'oferir comunicacions eficients.

En primer lloc, un dels seus usos més destacats, pel qual és conegut, és l'oferiment d'una xarxa global d'alta velocitat que a més a més de fer la internet més eficient, n'aporta a pobles petits, muntanyes o illes, regions remotes que mai han estat connectades per manca d'instal·lacions de comunicació.

En segon lloc, sovint les infraestructures de comunicació terrestres deixen de funcionar durant desastres naturals, això no passa amb els satèl·lits LEO, aquests permeten que els serveis d'emergència continuïn comunicats per poder gestionar la situació.

En tercer lloc, és capaç de proveir dades essencials per millorar les poblacions i apropar-les a la idea que tenim de les ciutats intel·ligents. Per exemple, en l'agricultura tradicional s'ha vist que fins al 30% d'aigua de reg s'evapora o es perd, i un 50% de fertilitzants contaminen els rius en lloc de fertilitzar la terra. Els satèl·lits poden recollir grans quantitats de dades sobre el sòl, temperatura, aigua, així com imatges de drons i estacions meteorològiques. Després es processaran per poder optimitzar el reg, millorar les collites i reduir l'impacte ambiental que en el món en què vivim on hi ha un nombre reduït de pagesos i molta demanda d'aliment, s'ha convertit en un problema amb molta urgència de resoldre. També disposen d'altres dades com de demogràfiques que permeten la planificació urbana, ja que s'ha detectat una tendència creixent d'urbanització.

3.5 Exemples actuals

Tot i ser una tecnologia més recent, els satèl·lits LEO han estat guanyant pes en el sector de les telecomunicacions. Actualment, les dues empreses que lideren el sector són:

- SpaceX: Va ser una de les primeres empreses a llançar una constel·lació LEO, anomenada Starlink, la qual s'ha convertit en la més gran de totes elles amb l'objectiu d'oferir internet d'alta velocitat i baixa latència als usuaris. L'any 2019 van

enviar els primers 60 satèl·lits que en formarien part, amb els anys han anat augmentant teixint una gran xarxa i actualment en pertanyen 7578, dels quals 7556 estan en funcionament. Prèviament, la majoria d'internet per satèl·lit provenia de les que tenien òrbita geoestacionària, que estan molt lluny de la terra. Això causava que les dades transmeses presentessin una latència alta perquè la informació tardava molt a anar i tornar entre l'usuari i el satèl·lit. Aquest retard provocava dificultats en algunes activitats determinades com fer videotrucades o veure vídeos en directe que necessiten una connexió ràpida. Starlink en ser una constel·lació d'òrbita baixa, permet que la informació es transmeti de manera molt més ràpida i millora l'execució d'activitats que necessiten baix retard.

- Oneweb: La constel·lació LEO anomenat Eutelsat Oneweb compta amb uns 648 satèl·lits distribuïts en 12 plans orbitals sincronitzats, a 1200 km altitud, amb els primers llançats l'any 2019. Solen ser enviats en grups cap a una òrbita d'espera inicial a una altitud d'aproximadament 450 km, des d'allà, ascendeixen fins a la seva altitud operativa (LEO) designada mitjançant propulsió elèctrica. Cada satèl·lit pot connectar-se de manera contínua amb les antenes i la xarxa terrestre a sota, transmetent dades en temps real i gran velocitat mentre es mouen. Les antenes a terra poden veure diversos satèl·lits a la vegada perquè es mouen ràpidament, aportant una connexió fiable i precisa, fins i tot en llocs on antigament no arribava bé internet.

D'altres sectors:

- Deimos-1 (monitoratge terrestre i gestió de riscos naturals): És el primer satèl·lit espanyol per l'observació de la Terra i per la prevenció i gestió en cas de desastres naturals que ha estat en funcionament des de 2009. A través de l'ús d'un sensor multiespectral proporciona imatges òptiques i infraroges adaptades a l'estudi de la coberta vegetal terrestre. Alguns dels seus projectes destacats seguits a través d'imatges del satèl·lit són:
 - Primeres imatges del terratrèmol i tsunami del Japó (2011)
 - Gran taca volcànica sobre l'illa del Hierro (2011)
 - Inundacions de l'Est d'Austràlia (2012)
 - L'incendi al Parc Natural del Teide a Tenerife (2012)
 - El gran incendi de la zona de la Jonquera a Girona (2012)

L'any 2012, amb només tres anys en funcionament, va aconseguir fer 15.900 voltes a la Terra i ha capturat unes 13.700 imatges. Dos anys després es va enviar

Deimos-2 que a diferència del primer, captura àrees més petites, però amb una resolució més alta. No són considerats una constel·lació, treballen de manera independent i tenen un objectiu complementari.

- LandSat 9 (recursos de la Terra): Va sorgir d'una col·laboració entre la NASA i la USGS (United States Geological Survey) que té com a missió controlar, entendre i gestionar recursos naturals per sostenir la vida humana. L'increment del canvi d'aquests darrers anys en l'ús del sòl i la naturalesa per a crear noves indústries i d'obtenció de materials a escala mundial té conseqüències profundes per al clima, el temps, l'economia i la salut de les persones. Landsat 9 observa la superfície terrestre i mostra els canvis produïts tant de manera natural com d'artificial. Forma la constel·lació LandSat amb LandSat-8. Són quasi idèntics i operen de manera coordinada.
- Terra (investigació): Antigament conegut com a EOS/AM-1 forma part d'un projecte d'observació de la Terra amb fins científics dut a terme pels Estats Units, Japó i Canadà. Recullen informació sobre diferents aspectes del nostre planeta utilitzant diversos instruments que van a bord com ASTER, CERES, MISR, MODIS I MOPITT que serveixen per observar propietat dels núvols, intercanvis d'energia carboni i aigua entre l'aire i la superfície terrestre i la presència de gasos contaminants a l'atmosfera. A partir d'aquestes dades es mesura en primer lloc, el sistema terrestre a escala global i durant tot l'any, incloent la productivitat dels ecosistemes. També es detecten millor l'impacte de les activitats humanes sobre el medi ambient per poder optimitzar els models que prediuen el canvi climàtic. D'aquesta manera s'aconsegueix començar un seguiment a llarg termini dels efectes de l'escalfament global arreu del món.

4. SISTEMES DE SEGUIMENT

4.1 Radar

Des del principi dels vols espacials, s'han utilitzat tècniques de radar per obtenir la següent informació:

- L'angle de punteria, la direcció en què cal apuntar l'antena per veure el satèl·lit. S'obté mesurant la direcció de l'amplitud màxima del senyal que emet. Es pot obtenir a través de dues formes:
 - Mètode d'exploració cònica: l'antena realitza unes lleugeres rotacions fins que cobreix la direcció del satèl·lit. Es guia mitjançant les variacions d'amplitud del

senyal rebut, si baixa vol dir que el feix s'ha allunyat i si augmenta vol dir que s'està acostant.

- Tècnica de monopuls: deriva els desplaçaments de l'angle de l'antena mitjançant l'extracció de dos senyals: la de diferència Δ que mostra quant t'has desviat i en quina direcció, i la de suma Σ , és el total i fa de referència. S'envien al sistema de seguiment i es produeixen sortides d'error que indicaran l'angle de punteria. Permet controlar automàticament el moviment de l'antena.
- La distància entre la nau i l'estació (ρ). El radar de mesurament bidireccional clàssic utilitza un senyal de mesurament que s'irradia des del centre de control de la Terra fins al satèl·lit. Un cop rebut, l'emet de tornada a l'estació terrestre. La darrera capta la transmissió de monitoratge i determina el seu temps de viatge τ . Això s'expressa com un valor de mesurament equivalent: $\rho = 1/2 \cdot c \cdot \tau$, on c és la velocitat de propagació del senyal electromagnètic. Dit d'una altra manera, és igual a la mitjana de la distància d'anada i tornada.
- Velocitat radial s'obté a partir de l'efecte Doppler d'una ona de radar emesa des de l'estació terrestre, transposada pel satèl·lit i rebuda de nou a la Terra. Es tracta d'un fenomen que passa amb qualsevol classe d'ona. Diu que la freqüència d'una ona canvia segons si la font de l'ona i l'observador s'acosten o s'allunyen. Augmenta quan la nau s'aproxima i disminueix quan es distancia. Això vol dir que la diferència entre la freqüència transmesa i la rebuda és directament proporcional a la velocitat radial respecte a l'estació.

El sistema pot ser unidireccional, en què el senyal només viatja en una direcció: del transmissor al receptor, o bidireccional, en dues (vegeu apartat 3.2). Pels d'òrbita baixa s'utilitzen generalment aquesta última mena, amb una estació terrestre que fa de transmissor i receptor. Es relaciona amb el fet que només uns quants satèl·lits estan equipats amb estàndards de temps i freqüència d'alta precisió per realitzar mesures unidireccionals. En el cas de les que utilitzen dues vies, l'equip d'alta precisió que ho permet està situat a l'estació terrestre.

4.2 Seguiment làser de satèl·lits (SLR)

És una tècnica molt més precisa que el radar per mesurar la distància entre una estació terrestre equipada amb un làser i un satèl·lit que porta retroreflectors que són petits miralls que reflecteixen la llum cap a l'emissor sense dispersió significativa. Consisteix en la

radiació d'un pols de llum làser cap a la nau que en arribar és rebotat. Quan retorna a la Terra és detectat pel telescopi de l'estació, que atura un cronòmetre d'alta precisió. Mesurant el temps d'anada i tornada i multiplicant-lo per la velocitat de la llum es calcula la distància exacta entre els dos elements.

Tanmateix, presenta algunes limitacions. No permet el seguiment automàtic, sinó que depèn de disposar d'òrbites prèvies molt precises per poder apuntar el telescopi. A més, el seu ús queda restringit per la dependència de les condicions meteorològiques. No es podrà dur a terme quan el cel estigui ennuvolat.

4.3 Fitxer TLE

Un fitxer TLE (Two-line element set) és un format de dades que consisteix en dues línies i que és essencial per rastrejar satèl·lits, ja que proveeixen una descripció concisa i detallada de l'òrbita mitjançant els seus elements orbitals que en el cas dels LEO es recopilen amb les tècniques anteriors. La primera línia indica elements orbitals i metadades que ajudaran a descriure els addicionals de la segona fila. Pot incloure una capçalera indicant el nom del satèl·lit, però és opcional. En cas que existís estaríem davant un "three-line element set", que s'abreuja de la mateixa manera: TLE.

Estan disponibles en diverses pàgines web com Space-Track o Planet. La que es farà servir serà Celestrak. On són derivats directament de dades proporcionades pels responsables o operadors.

4.3.1 Components

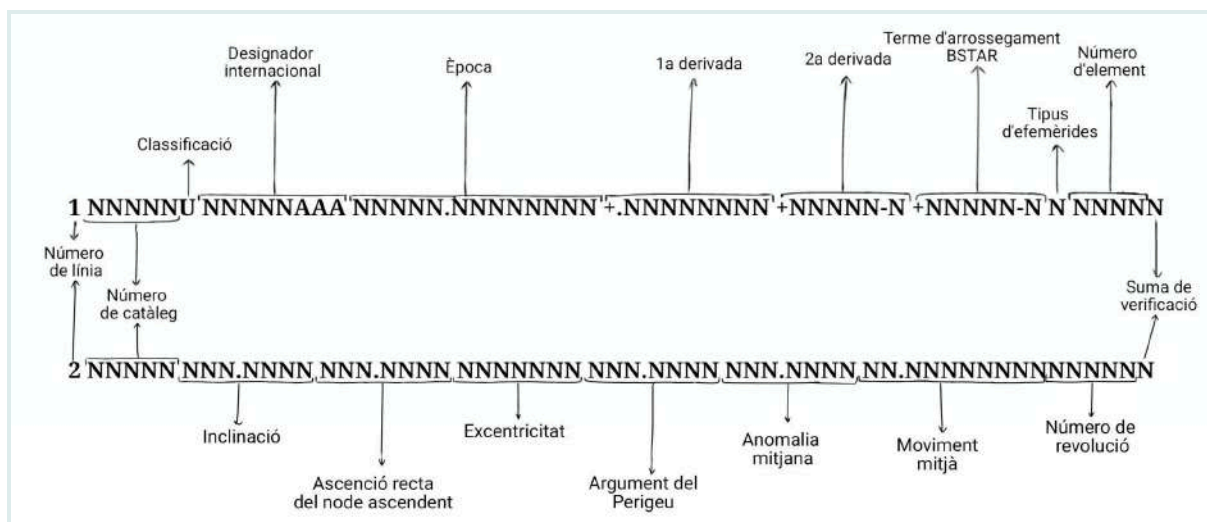


Fig. 8 Diferents parts d'un fitxer TLE
Font: pròpia

4.3.1.1 Número de línia

El primer dígit de cada línia del TLE (1 o 2) indica si és la primera o la segona fila.

4.3.1.2 Número de catàleg

Tots els objectes llançats a l'espai reben un identificador únic assignat pel Comandament Espacial dels Estats Units (USSPACECOM). El codi habitualment està format per cinc xifres que van del 00001 fins al 99999. No obstant això, s'ha introduït un nou sistema recentment, l'Alpha-5, on només s'utilitzen majúscules i números que amplia la numeració fins a un màxim de 339 999 elements. Les lletres "I" i "O" són descartades per evitar confusions amb els números "1" i "0".

Letter	Value	Letter	Value	Letter	Value
A	10	J	18	S	26
B	11	K	19	T	27
C	12	L	20	U	28
D	13	M	21	V	29
E	14	N	22	W	30
F	15	P	23	X	31
G	16	Q	24	Y	32
H	17	R	25	Z	33

Fig. 9 Sistema de catalogació Alpha-5

Font: <https://keeptrack.space/deep-dive/two-line-element-set>

Com més baix sigui el valor, més vell serà el satèl·lit.

4.3.1.3 Classificació

Cada TLE conté una lletra a la primera seqüència de números que indica si és pública: U (*unclassified*) i tothom hi pot accedir o secreta: S (secret) que és confidencial.

4.3.1.4 Designador internacional

Aporta informació sobre el llançament, segueix el format YYNNNP.

YY: són els dos últims dígits de l'any de l'enviament.

NNN: és el número d'ordre del llançament aquell any.

P: una o diverses lletres que indica quina peça s'ha enviat a l'espai. El caràcter A normalment s'utilitza per a la càrrega útil principal; és a dir, el satèl·lit o la peça principal de la missió, mentre que les altres són per components secundaris.

4.3.1.5 Època

Ofereix una marca de temps que revela quan es van registrar les dades orbitals. Segueix la següent estructura: AAD.H

AA: els dos últims dígit de l'any.

D: dia de l'any.

H: hora en fracció de dia en UTC. Per obtenir-la en el format que s'utilitza habitualment es multiplica per 24 per saber l'hora i aleshores els decimals són multiplicats per 60 amb els quals aconseguixes els minuts.

4.3.1.6 Primera i segona derivada del moviment mitjà

Mostren com varia la velocitat orbital amb el temps i si accelera o no. Juguen un paper important a l'hora de predir l'estat futur de l'òrbita del satèl·lit, sobretot en casos on es pot veure alterada per forces externes com la fricció atmosfèrica.

4.3.1.7 Terme d'arrossegament BSTAR

Estima els efectes de l'arrossegament atmosfèric sobre el moviment del satèl·lit.

4.3.1.8 Tipus efemèrides

Indica el model orbital utilitzat per generar les dades.

1=SGP

2=SGP4

3=SDP4

4=SGP8

5=SDP8

Tanmateix, a la pràctica, en els TLE públics el valor sempre serà 0, perquè només es distribueixen els generats amb SGP4 o SDP4. L'usuari ho haurà de saber per defecte.

4.3.1.9 Número d'element

Indica quina versió del conjunt de dades és. Incrementa cada cop que es genera un nou TLE per un satèl·lit.

4.3.1.10 Suma de verificació

Garanteix que les dades no hagin estat manipulades ni corrompudes. Per calcular-ho, se sumen els valors de tots els números de cada línia (ignorant totes les lletres, espais, punts i signes més) i s'assigna un valor d'1 a tots els signes menys. La suma de verificació és l'últim dígit d'aquesta operació. És un procediment molt senzill que pot arribar a detectar el 90% de tots els errors.

4.3.1.11 Inclinió

Vegeu apartat 1.2.2. Expressat en graus ($^{\circ}$), comprèn valors del 0 al 180.

4.3.1.12 Ascensió recta del node ascendent

Vegeu apartat 1.2.2. Mesurat en graus ($^{\circ}$), comprèn valors del 0 al 360.

4.3.1.13 Excentricitat

Vegeu apartat 1.2.1.1. No té unitats. Sempre s'expressa com un enter, però s'ha de llegir com si tingués un punt decimal a davant.

4.3.1.14 Argument Perigeu

Vegeu apartat 1.2.2. Es mesura en graus ($^{\circ}$) i comprèn valors del 0 al 360.

4.3.1.15 Anomalia mitjana

És la posició teòrica del satèl·lit respecte al perigeu en graus assumint que té un moviment uniforme per simplificar els càlculs. Mesurat en graus ($^{\circ}$).

4.3.1.16 Moviment mitjà

Expressat en revolucions per dia és la velocitat angular mitjana d'un cos en òrbita el·líptica.

4.3.1.17 Número de revolució

Ens informa de la quantitat de vegades que ha fet la volta a la Terra d'ençà que fou llançat.

PART PRÀCTICA

Aquí s'intentaran donar resposta a les hipòtesis formulades a la introducció del treball mitjançant el compliment de tots els objectius plantejats. S'ha estructurat en tres parts. En primer lloc, es crearà un codi basat en el simulador orbital SGP4 que permetrà la visualització de la trajectòria dels satèl·lits LEO en un mapa 2D. Els resultats obtinguts s'utilitzaran per determinar les similituds i diferències entre les diferents naus, que compleix el primer objectiu. En segon lloc, agafant una situació hipotètica en què es vol fer una planificació per a una observació aficionada des del punt de referència de l'institut Pere Alsius i processant la seva visibilitat en gràfics es podrà enllestir el segon. Finalment, es durà a terme l'últim que tracta de validar els fitxers TLE al llarg del temps. Es compararan les prediccions obtingudes a partir del fitxer del 25 de juliol amb els TLEs publicats en les dates futures corresponents.

Abans de res, s'ha decidit llegir el llibre *Introducció a la programació con Python*, ja que tot i que no és la primera vegada que es programa, només s'havien creat línies molt senzilles de codi amb aquesta llengua fa dos anys. El darrer es va treballar amb C++ per dur a terme alguns projectes amb plaques Arduino, pel que s'ha determinat que seria important revisar les funcions bàsiques de Python per facilitar la feina al moment d'analitzar el simulador orbital i codificar.

Per editar el codi s'ha emprat Visual Studio Code, una aplicació que amb un entorn amb més funcionalitats el fa més accessible per principiants. Disposa per exemple d'un sistema que et corregeix errors estilístics que en ser molt petits es poden passar per alt causant el mal funcionament del codi i un ressaltat de sintaxi que aplica colors diferents del text segons el tipus d'element que sigui.

5. SIMULADOR ORBITAL SGP4

SGP són les sigles de "Simplified General Perturbations", una sèrie de models matemàtics desenvolupats a partir dels anys 60 per a propagar la trajectòria de cossos que orbiten al voltant de la Terra. El 2006 es va publicar el codi font del programari que implementa el model SGP4, facilitant d'aquesta forma el càlcul de trajectòries orbitals a la comunitat científica.

5.1 Anàlisi del codi

El simulador orbital consta de quatre elements principals.

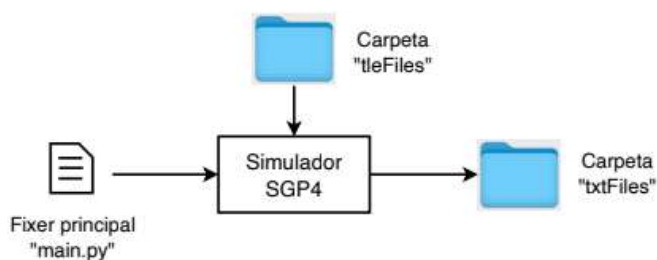


Fig. 10 Esquema del programa

Font: proporcionat pel tutor del programa Argó de la UAB

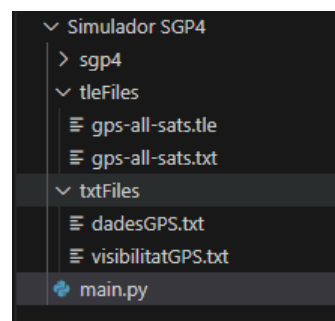


Fig. 11 Fitxers del simulador

Font: pròpia

La carpeta `tleFiles` ha de contenir el fitxer TLE amb els paràmetres orbitals de la constel·lació que es vol simular.

La carpeta `txtFiles` que, una vegada finalitzada la simulació, contindrà els fitxers de text amb els resultats. Se'n trobaran dos:

- `DadesGPS.txt`: on cada línia representa la posició d'un satèl·lit en un moment concret. A la primera columna hi trobem l'identificador, la segona indica la visibilitat, el valor "1" significa que és visible en aquell instant; d'altra banda, si apareix un 0, no ho és. La tercera, la quarta i la cinquena corresponen a la data, expressada en dia, mes i any. La sisena és l'hora exacta a la qual s'han obtingut les dades. La setena i la vuitena contenen els angles d'elevació i azimuth amb els quals és visible el satèl·lit. Les tres últimes contenen les coordenades X, Y i Z en el sistema ECEF (Earth-Centered Earth-Fixed) per descriure la posició respecte a la Terra.
- `VisibilitatGPS.txt`: La primera fila inclou la llista completa de totes les naus simulades. Les següents indiquen per a cada interval de temps quins es poden apreciar marcant amb un 1 a la columna corresponent als visibles.

`main.py` és el fitxer principal que cal executar per a fer córrer el simulador orbital i obtenir resultats. També és la que es treballarà principalment en la part pràctica (vegeu annex 4).

El codi comença definint els següents paràmetres:

- Radi de la Terra (`re`)
- Les coordenades del punt de referència on es troba l'usuari (`long_deg`, `lat_deg`)
- Les dates els quals es vol fer el càlcul (`data_any`, `data_mes`, `data_dia`, `data_hora`, `data_min`, `data_segons`)
- L'interval de temps per al qual volem calcular les òrbites (`num_days`, `num_hours`, `num_min`)
- L'elevació mínima que és l'angle que representa el límit inferior d'altura sobre l'horitzó a partir del qual el satèl·lit es considera visible (`min_elev`)

- Els noms dels fitxers d'entrada i sortida. La variable `t1e_file` serà el document que contindrà totes les dades TLE dels satèl·lits que posteriorment seran processats. `output_file` i `output-visFile` s'encarreguen de recollir les dades provinents dels càlculs, la primera emmagatzemarà la informació sobre la posició mentre que la darrera guardarà les de visibilitat.

Després, una sèrie de llibreries són carregades. Es tracta d'unes col·leccions de codi ja escrites per altres programadors que permeten fer tasques complexes, en aquest cas inclouen funcions per a la propagació orbital SGP4, càlculs d'efemèrides, transformacions de coordenades i funcions matemàtiques bàsiques com el sinus, cosinus, arccosinus, arctangent i la constant pi, així com la gestió de dates i hores.

```
37 import sys
38 from sgp4.earth_gravity import wgs84
39 from sgp4.io import twoline2rv
40 from sgp4.propagation import _gstime
41 from sgp4.ext import jday, eci_to_ecef, mag, dot
42 from datetime import datetime, date, time, timedelta
43 from math import (sin,cos,acos,atan,atan2,sqrt,pi)
```

Fig. 12 Importacions de llibreries
Font: pròpia

Per confirmar que no hi ha hagut cap problema durant aquest primer tram s'imprimeix un missatge a la terminal indicant que s'han començat a executar els càlculs.

La latitud i la longitud del punt de referència (`long_deg`, `lat_deg`) es transformen en radians amb l'equivalència: $180^\circ = \pi$ radians per a posteriori, obtenir les coordenades cartesianes X, Y i Z del punt de referència sobre la superfície terrestre amb la sèrie de fórmules del codi. La data i l'hora especificat anteriorment s'estructura en format ISO8601: YYYY-MM-DD HH:MM:SS. Els fitxers tant d'entrada com de sortida són oberts. Per una banda, el TLE és llegit i convertit en una tupla, una seqüència immutable de valors per assegurar que les dades no siguin accidentalment modificades. Addicionalment, ocupen menys espai. Per altra, els dos fitxers de sortida són preparats per escriure-hi les dades que generaran.

```
lon    = lon_deg*pi/180
lat    = lat_deg*pi/180

PX     = re*cos(lat)*cos(lon)
PY     = re*cos(lat)*sin(lon)
PZ     = re*sin(lat)
```

Fig. 13 Transformació de longitud i latitud en radians i obtenció de coordenades x, y, z
Font: pròpia

```
tlelines= tuple(open(tle_file,'r'))

out_file    = open(output_file,"w")
out_visFile = open(output_visFile,"w")
```

Fig. 14 Fitxers oberts preparats per ser llegits i per escriure-hi
Font: pròpia

Es calcula el vector que apunta des del centre de la Terra cap a la posició de l'observador amb les coordenades cartesianes i el normalitzem per aconseguir l'unitari.

A la variable `visible` es crea una matriu amb tantes columnes com satèl·lits hi ha i cada fila correspon a un moment concret seguint l'interval concretat anteriorment. Per fer-ho primer es determinen els instants totals multiplicant els dies, les hores i els intervals de minut durant els quals volem fer la predicció. Després s'inicialitza un bucle "for", que recorre una seqüència d'elements i executa un bloc de codi per a cada component. En aquest cas, se li ha indicat que escrigui un 0 per cadascun d'aquests instants. Tot seguit, se li aplica un segon bucle que repeteix el procés anterior pel nombre de satèl·lits que s'obté fent la divisió entre tres de la variable `tlelines`, que prèviament hem definit com la quantitat total de línies del fitxer TLE.

```
visible = [[0 for x in range(num_days*num_hours*num_min)] for x in range(len(tlelines)//3)]
```

Fig. 15 Variable `visible`
Font: pròpia

La nova variable `prn` s'encarrega de crear una fila amb tants 0 com nombre de satèl·lits hi ha que posteriorment seran reemplaçats pels identificadors.

```
prn      = [0 for x in range(len(tlelines)//3)]
```

Fig. 16 Variable `prn`
Font: pròpia

La part principal del programa és un bucle que recorre els satèl·lits del fitxer TLE. Llegeix les línies de tres en tres i s'extreu el seu identificador. Es crea un objecte que permet calcular la seva posició en el temps i s'inicialitzen variables per controlar l'ordre dels càlculs i controlar la identificació de la nau. A dintre s'executa tres nivells més de bucles que recorren tots els dies, hores i minuts de l'interval temporal definit. Es fa servir `timedelta` per generar cada moment exacte del període. S'encarrega de calcular-ho sumant temps a la variable `datetime`. Aleshores es fa la propagació del satèl·lit mitjançant el mètode `.propagate()` de la llibreria SGP4 que calcula la posició i la velocitat del satèl·lit en coordenades inercials ECI, les quals les x, y, z, són fixes respecte a estrelles llunyanes, no giren amb la Terra.

Després es converteixen en coordenades d'un sistema fix (ECEF) on estan alineades amb la superfície terrestre. S'aconsegueix calculant primer el dia julià perquè facilita els càlculs astronòmics. Es tracta d'un comptador de dies i fraccions des de l'any 4713 aC. La funció `_gstime()` obté el temps sideral de Greenwich, el nombre de graus que has de girar el globus perquè la longitud de Greenwich (0°) coincideixi amb la posició correcta respecte a les estrelles. Aquest valor és necessari per a la transformació que es fa amb `eci_to_ecef` que prové de la llibreria SGP4. Per l'obtenció del seu vector, amb sentit a la nau, s'agafa les coordenades del satèl·lit que són el resultat obtingut del procés anterior de l'observador i se li resta la posició de l'observador. És normalitzat per obtenir l'unitari que té la mateixa direcció, però amb longitud igual a 1.

A continuació, es defineixen les direccions nord i est respecte a la posició de l'usuari a partir de les coordenades geogràfiques de l'observador i després es normalitzen. Gràcies a això, es poden obtenir els components necessaris per determinar la direcció horitzontal en què cal mirar des d'un punt concret de la Terra per veure el satèl·lit que es mesura amb graus. Va de 0° a 360°, començant des del nord (0°), passant per l'est (90°), el sud (180°) i l'oest (270°) i conegut com a azimuth. Ara tenim els vectors unitaris, per obtenir les projeccions es faran els productes escalars del `PEarth2Sat_norm` sobre el del nord: `North_norm` i el de l'est: `East_norm` amb la funció `dot()`. És una operació matemàtica que segueix l'estructura s'aplica a dos vectors i produeix un escalar (un nombre) com a resultat.

Aleshores, s'aplica la seva fórmula: $\text{azimut} = \text{atan2}(\text{projecció sobre Est}, \text{projecció sobre Nord})$ i es multiplica pel factor $180/\pi$ per obtenir el resultat en graus. `atan2` és una variant de l'arctangent, a diferència de l'habitual té en compte el signe de tots dos arguments i permet identificar correctament el quadrant en què es troba. Ara bé, per defecte retorna valors dins l'interval $[-180^\circ, 180^\circ]$. Tot i ser correcte matemàticament, per la seva representació no és pràctic, ja que aquest es defineix sempre dintre el rang $[0^\circ, 360^\circ]$. Per això quan el resultat surt negatiu se li suma 360° per transformar-lo en el seu equivalent positiu.

```
# Calcular l'angle d'azimuth
azi_angle_deg = atan2(dot(PEarth2Sat_norm,East_norm),dot(PEarth2Sat_norm,North_norm))*180/pi

# Mantenir l'azimuth dintre el rang [0,360) graus
if azi_angle_deg<0:
    azi_angle_deg = azi_angle_deg+360
```

Fig. 17 Càlcul de l'angle d'azimuth i adaptació del resultat a les nostres necessitats
Font: pròpia

L'elevació s'aconsegueix calculant l'angle entre la vertical de l'observador i el vector que va cap al satèl·lit per obtenir `off_vert_angle_deg` i restar-ho a 90°. Així doncs, una elevació de 0° correspon a l'horitzó i 90° al zenit. Si `off_vert_angle_deg` és menor a 90° i supera

un mínim predefinit, que per defecte és l'horitzó, es marca com a visible. Addicionalment, s'imprimeixen totes les dades en el fitxer `dadesGPS`. En cas contrari, només s'escriu la data.

Finalment, recorre a la matriu i a la fila que hem creat abans anomenades `prn` i `visible` per escriure un fitxer resum per les visibilitats. A la primera línia s'escriuen els identificadors i després, per cada instant, s'escriu una línia amb 1 o 0 indicant si era visible o no en aquell moment. Quan ha acabat, es tanca el fitxer amb els TLEs i s'imprimeix "Finalitzat" a la terminal perquè l'usuari ho sàpiga.

Dins de la carpeta SGP4 hi ha tot el codi pròpiament del simulador SGP4 que en aquest cas serà com una caixa forta on no caldrà modificar res.

Està format per fitxers de codi que junts componen:

- L'estat orbital: representa la posició i velocitat del satèl·lit, gestionat per `io.py` que s'encarrega de transformar un TLE en l'estat orbital inicial que utilitzarà el propagador i estructurat amb models de gravetat i constants definides per: `earth_gravity.py` i `model.py`.
- El propagador SGP4: calcula la posició i la velocitat en qualsevol instant futur a partir de l'estat inicial, implementat a `propagation.py` i recolzat per funcions matemàtiques auxiliars `ext.py`.

`__init__.py` simplement marca el directori com un mòdul Python i `tests.py` conté proves per assegurar el bon funcionament del simulador.

5.2. Adaptació del simulador

Per defecte, la versió del simulador SGP4 proporcionat està pensada per a processar fitxers TLE de la constel·lació de satèl·lits de GPS. Tot i que aquesta constel·lació no és LEO, sinó MEO, i informa sobre la visibilitat a la Universitat Autònoma de Barcelona. Però per la part pràctica no es busquen obtenir dades d'aquest tipus de satèl·lit ni d'aquest lloc, per tant, s'ha d'adaptar a les necessitats que es tenen.

5.2.1 Configuració de paràmetres

5.2.1.1 Punt de referència

Per començar el punt de referència que es necessita és l'institut Pere Alsius i Torrent de Banyoles. Llavors s'ha utilitzat Google Earth per obtenir la latitud i longitud en graus.

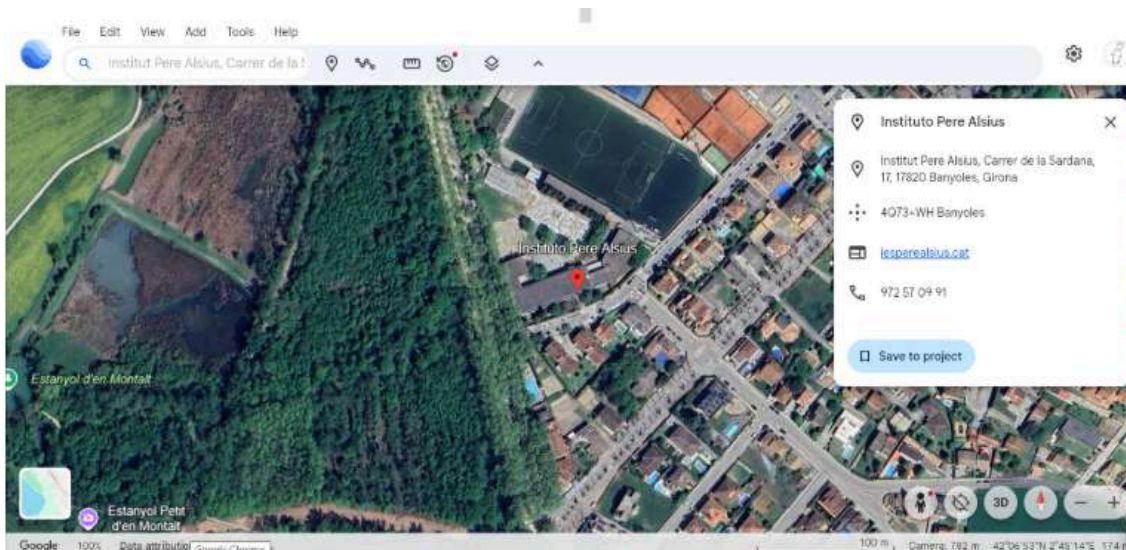


Fig. 18 Institut Pere Alsius i Torrent a Google Earth

Font: <https://earth.google.com/web/search/Institut+Pere+Alsius,+Carrer+de+la+Sardana,+Banyoles/>



Fig. 19 Coordenades de l'institut Pere Alsius i Torrent des de Google Earth

Font: <https://earth.google.com/web/search/Institut+Pere+Alsius,+Carrer+de+la+Sardana,+Banyoles/>

Com que el valor donat era en format complex, és a dir que es dividia en graus(°), minuts(') i segons("), i el simulador utilitza la incomplexa on s'expressa tot en graus, s'ha hagut de fer una conversió d'unitats.

Per la latitud: 42°06'53"

El 42, en estar ja en graus es manté igual. Els minuts, en canvi, es multipliquen pel factor de conversió 1°/60', pel fet que 1 grau equival a 60 minuts.

$$6' \times \frac{1^\circ}{60'} = 0,1^\circ$$

Amb els segons es fa el mateix producte, però amb el seu respectiu factor 1°/3600":

$$53'' \times \frac{1^\circ}{3600''} = 0,01472^\circ$$

Finalment, se sumen els graus inicials i els resultats anteriors, els minuts i segons que s'han transformat i s'arrodoneixen a la cinquena xifra decimal tal com ho fa el simulador.

$$42^\circ + 0,1^\circ + 0,01583^\circ = 42,11472^\circ = 42,11472^\circ$$

Ara es repeteix el procés amb la longitud: 2°45'14"

$$45' \times \frac{1^\circ}{60'} = 0,75^\circ$$

$$14'' \times \frac{1^\circ}{3600'} = 0,0038^\circ$$

$$2^\circ + 0,75^\circ + 0,0038^\circ = 2,7538^\circ = 2,75389^\circ$$

Els resultats s'introdueixen a les variables `lon_deg` i `lat_deg` del codi.

```

4
5 # Coordenades del punt de referència on es troba l'usuari observador
6 lon_deg = 2.75389
7 lat_deg = 42.1147
8

```

Fig. 20 Modificant les coordenades per les obtingudes
Font: pròpia

5.2.1.2 Modificació del text imprès

No obstant això, quan s'imprimeix amb `print()`, tot i que ens dona les coordenades correctes, segueix indicant que és la ubicació de la UAB. Per tant, s'ha anat a la línia de codi 58 on s'ha modificat "Coordenades Bellaterra UAB" per "Coordenades Institut Pere Alsius i Torrent de Banyoles".

```

58 print(" Coordenades Bellaterra UAB\n Long: %12.7f (deg)\n Lat: %13.7f (deg)"% (lon_deg,lat_deg))
59 print(" X: %15.7f\n Y: %15.7f\n Z: %15.7f\n" % (PX,PY,PZ))

```

Fig. 21 Modificació de la línia de codi per obtenir el text imprès actualitzat
Font: pròpia

Una vegada canviat, s'ha pogut comprovar que imprimeix el nom del lloc correcte.

```

58 print(" Coordenades Institut Pere Alsius i Torrent de Banyoles\n Long: %12.7f (deg)\n Lat: %13.7f (deg)"% (lon_deg,lat_deg))
59 print(" X: %15.7f\n Y: %15.7f\n Z: %15.7f\n" % (PX,PY,PZ))

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

> & C:/Users/LinaGuanCai/AppData/Local/Microsoft/WindowsApps/python3.11.exe
 "c:/Users/LinaGuanCai/Downloads/Simulador SGP4/main.py"
 Executant model orbital SGP4 amb Python...
 Coordenades Institut Pere Alsius i Torrent de Banyoles
 Long: 2.7538900 (deg)
 Lat: 42.1147000 (deg)
 X: 4720.5741033
 Y: 227.0666700
 Z: 4272.5006568
 Finalitzat!
 PS C:\Users\LinaGuanCai\Downloads\Simulador SGP4>

Fig. 22 Comprovació del text imprès
Font: pròpia

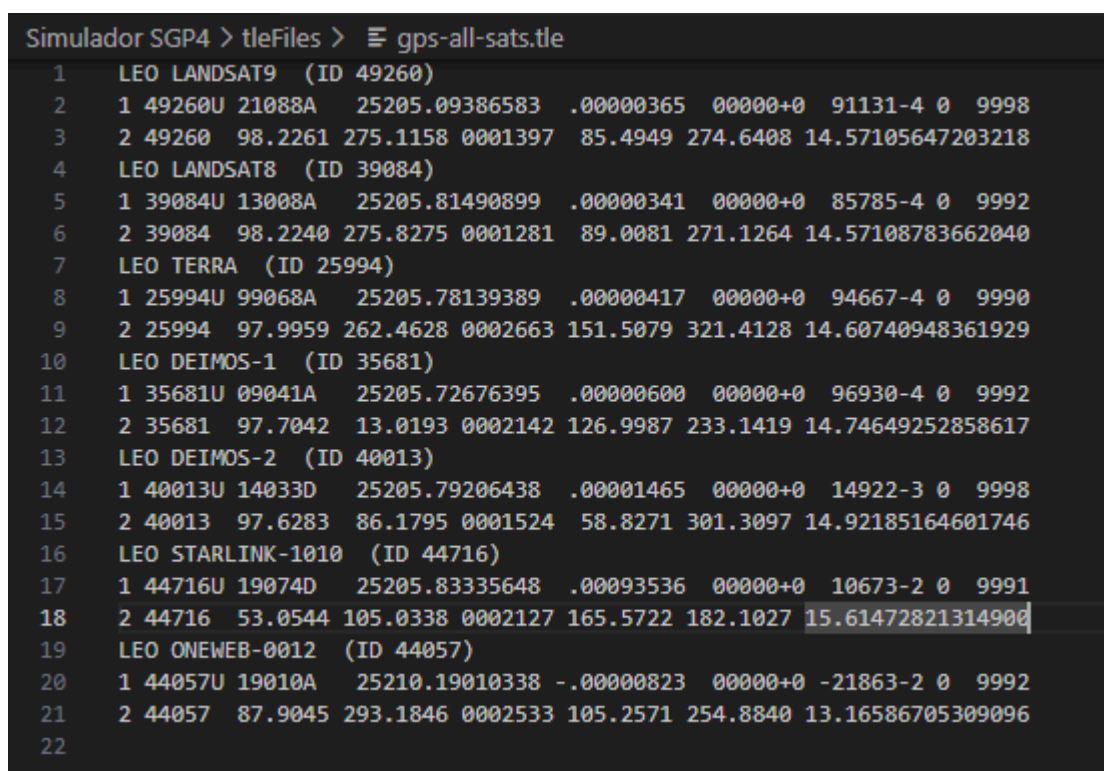
5.2.1.3 Data

Es canvien les variables `data_any`, `data_mes`, `data_dia` per la data en què es vol començar la simulació, en aquest cas, el 25 de juliol del 2025 que és el dia que es va adaptar el programa. I a `num_days`, `num_hours`, `num_min` s'hi introduirà l'interval de temps durant el qual es farà la predicció.

5.2.2 Satèl·lits per simular

Durant la recerca d'informació en l'apartat 3, s'ha vist que els satèl·lits LEO poden assumir funcions molt diverses. Per aquest motiu a l'hora de posar els exemples actuals (apartat 3.5), s'han intentat escollir els més importants de cada sector. És per això que per la part pràctica se simularan els que s'han seleccionat abans.

L'objectiu és figurar satèl·lits LEO, en comptes de MEO, així que s'ha accedit al fitxer `gps-all-sats.tle`, s'han esborrat les dades originals i s'han substituït per les que interessen, extretes de la pàgina web Celestrak. Per satèl·lits que formen part de constel·lacions petites com LandSat-9 que només està constituïda per dues naus, s'han inclòs completes. No obstant això, en el cas de Starlink o Oneweb que són més de 500 s'ha decidit utilitzar únicament el més vell que es mantingui en òrbita.



Simulador SGP4 > tleFiles > gps-all-sats.tle										
1	LEO LANDSAT9 (ID 49260)									
2	1	49260U	21088A	25205.09386583	.00000365	00000+0	91131-4	0	9998	
3	2	49260	98.2261	275.1158	0001397	85.4949	274.6408	14.57105647203218		
4	LEO LANDSAT8 (ID 39084)									
5	1	39084U	13008A	25205.81490899	.00000341	00000+0	85785-4	0	9992	
6	2	39084	98.2240	275.8275	0001281	89.0081	271.1264	14.57108783662040		
7	LEO TERRA (ID 25994)									
8	1	25994U	99068A	25205.78139389	.00000417	00000+0	94667-4	0	9990	
9	2	25994	97.9959	262.4628	0002663	151.5079	321.4128	14.60740948361929		
10	LEO DEIMOS-1 (ID 35681)									
11	1	35681U	09041A	25205.72676395	.00000600	00000+0	96930-4	0	9992	
12	2	35681	97.7042	13.0193	0002142	126.9987	233.1419	14.74649252858617		
13	LEO DEIMOS-2 (ID 40013)									
14	1	40013U	14033D	25205.79206438	.00001465	00000+0	14922-3	0	9998	
15	2	40013	97.6283	86.1795	0001524	58.8271	301.3097	14.92185164601746		
16	LEO STARLINK-1010 (ID 44716)									
17	1	44716U	19074D	25205.83335648	.00093536	00000+0	10673-2	0	9991	
18	2	44716	53.0544	105.0338	0002127	165.5722	182.1027	15.61472821314900		
19	LEO ONEWEB-0012 (ID 44057)									
20	1	44057U	19010A	25210.19010338	-.00000823	00000+0	-21863-2	0	9992	
21	2	44057	87.9045	293.1846	0002533	105.2571	254.8840	13.16586705309096		
22										

Fig. 23 Fitxers TLE dels satèl·lits LEO que se simularan
Font: pròpia

En aquest moment, el codi només imprimeix la posició del satèl·lit, l'angle d'elevació i l'angle d'azimut quan és visible; en altres paraules, quan l'elevació supera un cert mínim. Això impedeix fer una representació total de la trajectòria, ja que no retorna la informació essencial per la seva realització quan està per sota de l'horitzó. Per fer-ho possible, s'ha modificat el codi perquè imprimeixi les dades completes en tots dos casos.

Per començar, s'ha extret `elev_angle=90-off_vert_angle_deg` fora del bucle perquè es faci sempre i estigui disponible, independentment de la visibilitat.

Aleshores, s'ha modificat la instrucció `out_file.write()` escrivint el mateix que contenia la línia dintre de `if off_vert_angle_deg<=90` i provoca que imprimeixi la posició, l'angle d'elevació i d'azimut tant si és visible o no.

Perquè la informació de sortida sigui més entenedora el format de l'hora impresa fou canviada. Al principi, els minuts s'escriuen com un sol enter, circumstància que podria ocasionar problemes quan vols saber l'hora completa. Canviant `%2.0f:%1.0f` per `%02d:%02d` fa que es mostrin dos dígit.

```
off_vert_angle_deg = acos(dot(PEarth2Sat_norm,U_norm))*180/pi
elev_angle= 90-off_vert_angle_deg
if off_vert_angle_deg<= 90:
    if elev_angle >= min_elev:
        visible[sat_index][time_index] = 1
        out_file.write("%s %d %2.0f %2.0f %4.0f %02d:%02d %5.1f (deg) %5.1f (deg) %15.7f %15.7f %15.7f\n" % (line0,v
        #print("PRN%s %d %2.0f %2.0f %4.0f _ %2.0f:%1.0f _ %5.1f (deg) %5.1f (deg) %15.7f %15.7f %15.7f" % (line0,v
    # En cas que el satèl.lit no sigui visible simplement escrivim la data, pero no l'angle d'elevació
    else:
        visible[sat_index][time_index] = 0
        out_file.write("%s %d %2.0f %2.0f %4.0f %02d:%02d %5.1f (deg) %5.1f (deg) %15.7f %15.7f %15.7f\n" % (line0,v
        #print("PRN%s %d %2.0f %2.0f %4.0f _ %2.0f:%1.0f" % (line0,visible[sat_index][time_index],dt.day,dt.month,dt
```

Fig. 24 Canvis realitzats
Font: pròpia

6. CREACIÓ DE CODI PER VISUALITZAR LA TRAJECTÒRIA DELS SATÈL·LITS

6.1 Recerca d'informació

Mai havia fet una animació gràfica pel que s'ha hagut de fer una recerca d'informació sobre com començar. Primer es va plantejar quina aplicació es faria servir. Tot i que va cridar l'atenció "Matlab", una aplicació considerada molt bona i intuïtiva en ser usada per moltes universitats, es va optar per continuar amb Python en l'aplicació Visual Studio Code, perquè el primer utilitzava el llenguatge de programació C++ o C, i jo estava més familiaritzada amb l'anterior.

També es van visitar les documentacions oficials de les llibreries que es volien instal·lar (vegeu apartat 6.2) per fer una ullada en què consistien. A l'hora de programar totes aquestes van ser de gran ajuda. Ofereixen exemples, tutorials i explicacions detallades de les funcions disponibles, fet que facilita molt entendre com operen i resoldre qualsevol dubte que ha sorgit.

6.2 Descarregar llibreries

Per dur-ho a terme es van haver de descarregar algunes llibreries. Són col·leccions de codis que contenen funcions que pots utilitzar a l'hora de programar. Facilita la feina, ja que d'aquesta manera no s'ha de programar tot des de zero, procés que ningú fa. En aquest cas, es van haver de descarregar les següents:

- Numpy: permet fer càlculs numèrics i científics.
- Matplotlib: conté funcions que generen gràfics per visualitzar dades de diverses formes: gràfics, histogrames, entre altres. Serveix per representar resultats.
- Cartopy: està dissenyat per al processament de dades geoespacial amb l'objectiu de generar mapes i altres anàlisis relacionades amb dades geogràfiques.

Es va baixar a través de la terminal mitjançant el comandament: "pip install".

No obstant això, no va ser tan fàcil. El primer error va sorgir. Segons el contingut del missatge enviat pel sistema davant d'aquest problema, comentava que l'ordinador no tenia la compatibilitat amb Windows Long Path activat. En altres paraules, la longitud màxima d'una ruta són 260 caràcters, però s'ha inclòs la instal·lació de programes que superen aquest límit. Per arreglar-ho es va intentar activar-lo des del registre del sistema. Tanmateix, no es va aconseguir, ja que requerien permisos d'administrador que utilitzant el portàtil de l'institut no tenia. Es va decantar per crear un espai virtual, una còpia autònoma de Python per a un sol projecte, amb les seves pròpies biblioteques i configuracions. Manté el projecte aïllat del Python global. L'ús d'aquest recurs ha permès resoldre el problema perquè fa les descàrregues en una carpeta dins del projecte i la seva ubicació sol ser molt més curta que la ruta predeterminada que utilitza Python, així assegurem que no superem els 260 caràcters.

Es va crear escrivint la següent ordre a la terminal, on el darrer mot correspon al nom de l'entorn:

```
python -m venv venv
```

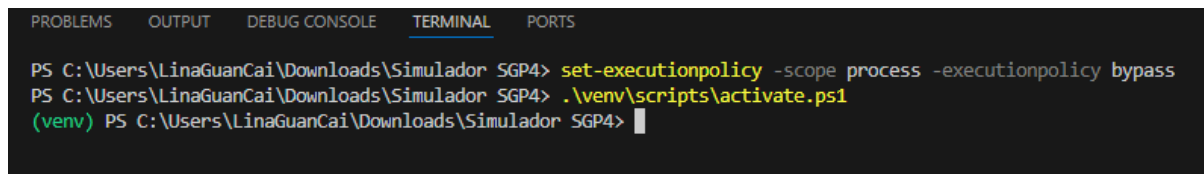
Per activar-lo es va fer:

```
.\venv\Scripts\Activate.ps1
```

El primer intent no va funcionar. Mostrava un error. Indicava que l'execució de l'*script*² estava deshabilitada. La raó darrere aquest esdeveniment és que la Powershell³ té aquesta mesura de seguretat que impedeix la seva execució per evitar que es posi en marxa un codi maliciós sense el consentiment de l'usuari. Hi havia dues opcions per concloure l'error: la primera era habilitar-ho, que amb la suspensió de la protecció del Powershell davant amenaces podria posar en perill l'ordinador; la segona era modificar la política només per l'actual sessió, és a dir, una vegada tancada tornar a la configuració original, amb les mesures de seguretat activades. Per garantir que el dispositiu no estigués sotmès a cap virus, es va decantar per l'opció número dos que és aconseguir indicant a la terminal:

Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass

Després es va tornar a activar. Es pot comprovar que s'ha assolit correctament en aparèixer el nom de l'entorn en color verd davant de la ubicació del fitxer:



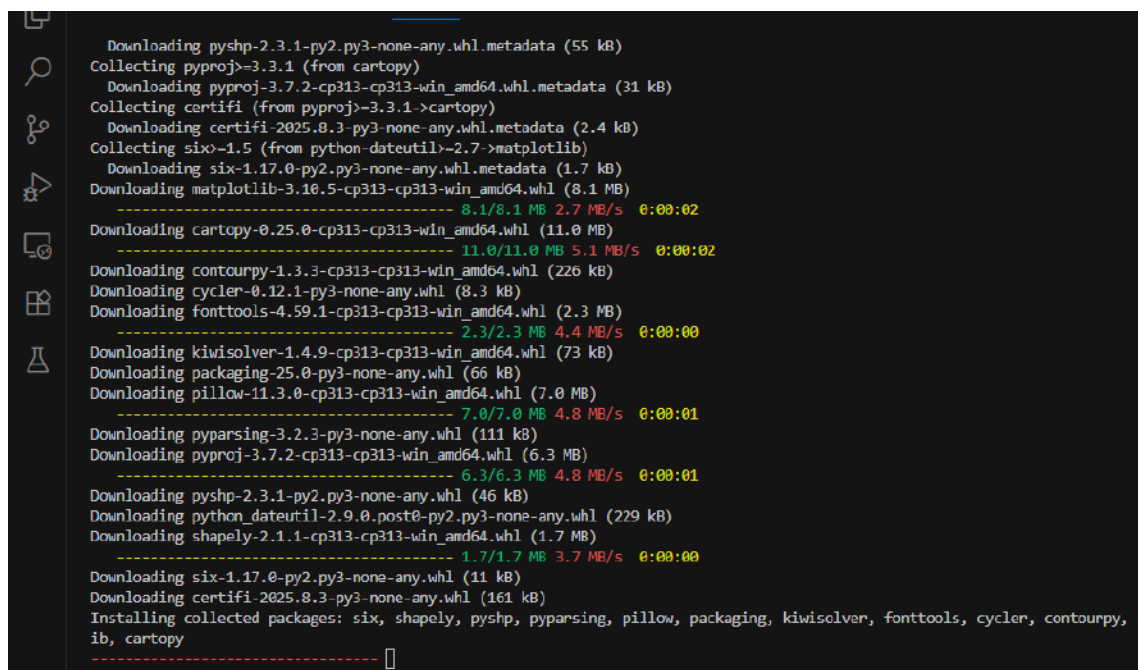
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\LinaGuanCai\Downloads\Simulador SGP4> set-executionpolicy -scope process -executionpolicy bypass
PS C:\Users\LinaGuanCai\Downloads\Simulador SGP4> .\venv\scripts\activate.ps1
(venv) PS C:\Users\LinaGuanCai\Downloads\Simulador SGP4>

```

Fig. 25 Activació de l'entorn virtual "venv"
Font: pròpia

Tot seguit també es van poder descarregar les llibreries.



```

Downloading pyshp-2.3.1-py2.py3-none-any.whl.metadata (55 kB)
Collecting pyproj==3.3.1 (from cartopy)
Downloading pyproj-3.7.2-cp313-cp313-win_amd64.whl.metadata (31 kB)
Collecting certifi (from pyproj==3.3.1->cartopy)
Downloading certifi-2025.8.3-py3-none-any.whl.metadata (2.4 kB)
Collecting six>=1.5 (from python-dateutil>=2.7->matplotlib)
Downloading six-1.17.0-py2.py3-none-any.whl.metadata (1.7 kB)
Downloading matplotlib-3.10.5-cp313-cp313-win_amd64.whl (8.1 MB)
----- 8.1/8.1 MB 2.7 MB/s 0:00:02
Downloading cartopy-0.25.0-cp313-cp313-win_amd64.whl (11.0 MB)
----- 11.0/11.0 MB 5.1 MB/s 0:00:02
Downloading contourpy-1.3.3-cp313-cp313-win_amd64.whl (226 kB)
Downloading cyclor-0.12.1-py3-none-any.whl (8.3 kB)
Downloading fonttools-4.59.1-cp313-cp313-win_amd64.whl (2.3 MB)
----- 2.3/2.3 MB 4.4 MB/s 0:00:00
Downloading kiwisolver-1.4.9-cp313-cp313-win_amd64.whl (73 kB)
Downloading packaging-25.0-py3-none-any.whl (66 kB)
Downloading pillow-11.3.0-cp313-cp313-win_amd64.whl (7.0 MB)
----- 7.0/7.0 MB 4.8 MB/s 0:00:01
Downloading pyparsing-3.2.3-py3-none-any.whl (111 kB)
Downloading pyproj-3.7.2-cp313-cp313-win_amd64.whl (6.3 MB)
----- 6.3/6.3 MB 4.8 MB/s 0:00:01
Downloading pyshp-2.3.1-py2.py3-none-any.whl (46 kB)
Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
Downloading shapely-2.1.1-cp313-cp313-win_amd64.whl (1.7 MB)
----- 1.7/1.7 MB 3.7 MB/s 0:00:00
Downloading six-1.17.0-py2.py3-none-any.whl (11 kB)
Downloading certifi-2025.8.3-py3-none-any.whl (161 kB)
Installing collected packages: six, shapely, pyshp, pyparsing, pillow, packaging, kiwisolver, fonttools, cyclor, contourpy,
ib, cartopy
-----

```

Fig. 26 Descarregant llibreries
Font: pròpia

² *scripts*: conjunt d'ordres escrites en un llenguatge de programació que s'executen de manera seqüencial per automatitzar una tasca.

³ Powershell: eina que permet donar instruccions a través de comandaments.

6.3 Primera versió

Es va començar fent una primera versió que es basava en un bucle “for” que es dedicava a llegir el fitxer que contenia les dades i representar-les línia per línia sobre el mapa. No es va fer servir cap mètode d'animació formal; però, tot i això, la representació s'actualitzava de manera generativa i feia que provoqués una sensació de moviment. Com a resultat, es podria considerar una animació visualment.

6.3.1 Importació de llibreries

El primer que s'ha fet és importar les llibreries que s'utilitzaran amb la instrucció `import`. Ja estan descarregades, però si no es fa només romandran a dintre l'emmagatzematge de l'ordinador, el codi no els podrà utilitzar en l'*script* concret.

```
import numpy
import matplotlib.pyplot as plt
import cartopy.crs as eix
import cartopy.feature as feature
```

Tot i que no és necessari, s'ha fet “`import (nom de la llibreria) as (àlies)`” que canvia el nom de la llibreria per un altre, generalment més curt per comoditat. Per exemple, un cop realitzat, ja no caldrà redactar “`matplotlib.pyplot`” cada vegada que es faci servir una de les seves funcions, es podrà simplement escriure “`plt`”.

6.3.2 Configuració del mapa 2D

El següent pas ha estat preparar l'espai on es representaran les dades. Inspirant-se per exemples i instruccions de la pàgina oficial de Matplotlib es crea una nova figura amb una mida de 12 polzades d'ample per 6 d'altura. `Figura` serà l'espai total on es poden dibuixar els gràfics i escriure elements. `gràfic` forma un subespai dintre de la figura, on es dibuixarà el gràfic en si, que quan li passes `projection=ccrs.PlateCarree()` els eixos s'adaptaran als que utilitza la projecció cartogràfica Plate Carrée on l'horitzontal correspon a la longitud i la vertical a la latitud. També s'han de definir els límits de `gràfic`. S'indica que horitzontalment anirà de -180° a 180° i verticalment de -90° a 90°, sense això inicialment el mapa s'anava reajustant de manera automàtica mentre es dibuixava.

```
figura = plt.figure(figsize=(12,6))
gràfic = plt.axes(projection=eix.PlateCarree())
gràfic.set_extent([-180, 180, -90, 90])
```

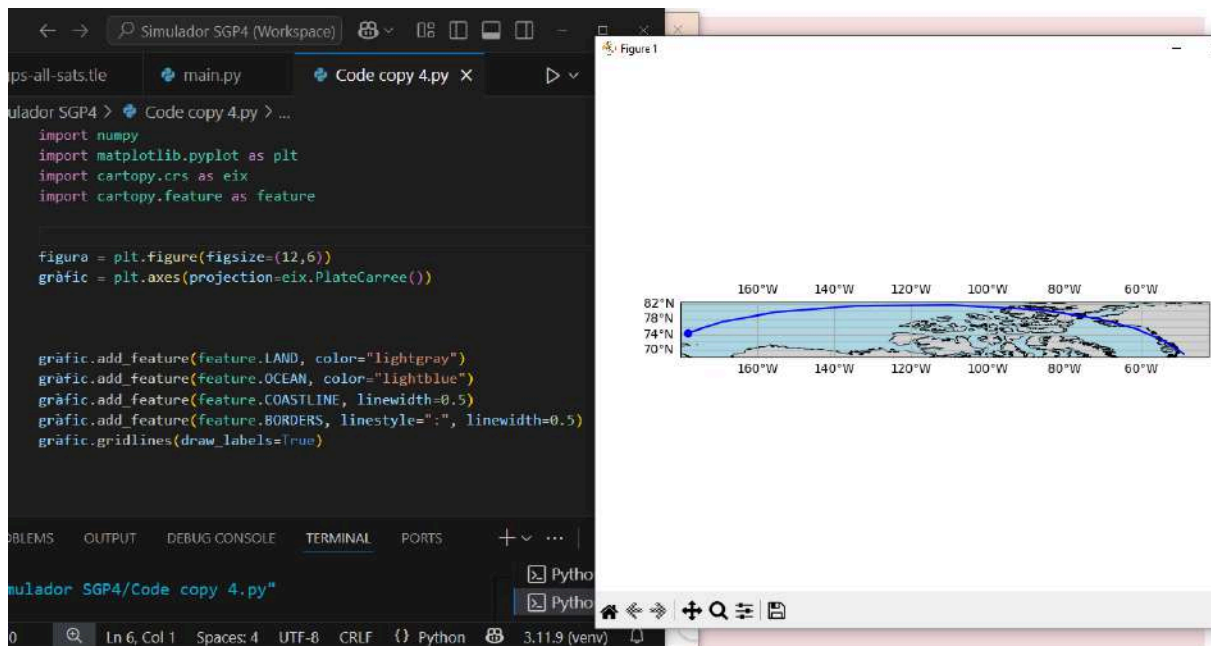


Fig. 27 Reajust automàtic si no fas servir `ax.set_extent`
Font: pròpia

S'afegeixen les característiques del planisferi com els oceans, els continents, les línies que delimiten les costes i les fronteres entre països. Es duu a terme amb la funció `ax.add_feature()` que permet a la vegada personalitzar-lo, dona llibertat de triar els colors dels elements acabats de mencionar i el tipus de línies. A més a més, per ajudar l'usuari a interpretar on es troben els punts s'han afegit línies de graella que corresponen a la longitud i la latitud amb els seus valors numèrics als marges.

```
gràfic.add_feature(feature.LAND, color="lightgray")
gràfic.add_feature(feature.OCEAN, color="lightblue")
gràfic.add_feature(feature.COASTLINE, linewidth=0.5)
gràfic.add_feature(feature.BORDERS, linestyle=":", linewidth=0.5)
gràfic.gridlines(draw_labels=True)
```

6.3.3 Transformació de coordenades ECEF a latitud i longitud

S'ha definit les funcions `latitud_calcular(x, y, z)` i `longitud_calcular(x,y)` per transformar les coordenades ECEF a latitud i longitud perquè es treballa en 2D. Al principi s'havien partit de les fórmules inicials utilitzades pel simulador orbital:

```
PX = re*cos(latitud)*cos(longitud)
PY = re*cos(latitud)*sin(longitud)
PZ = re*sin(latitud)
```

I s'havien aïllat amb la finalitat d'obtenir la longitud i latitud:

$$lat = \arcsin\left(\frac{z}{re}\right)$$

$$lon = \arcsin\left(\frac{y}{re \cdot \cos(lat)}\right)$$

Va resultar com a erroni. Fou causat pel fet que en el simulador orbital es fa una conversió directa de latitud i longitud a coordenades cartesianes ECEF per a un punt situat a la superfície terrestre, assumint que el radi és constant i igual al de la Terra. És una afirmació certa en aquell cas, però el raonament que no és correcte quan s'intenta invertir el procés a partir d'un punt fora de la superfície. En conseqüència, en la fórmula de la latitud z sempre serà major a r_e , així doncs la divisió serà més gran que 1 i quan fas sinus ha d'estar comprès entre -1 i 1. Ha estat un fet que ha causat problemes en totes dues equacions, ja que una es basa en el resultat de l'altra. Després es va recórrer a les següents, tal com indica Tonfilm (2015):

$$radius = \sqrt{(x^2 + y^2 + z^2)}$$

$$latitude = \arcsin\left(\frac{z}{radius}\right)$$

$$longitude = \text{atan2}(y, x)^{4}$$

La primera fórmula calcula la distància real entre el satèl·lit i el centre del planeta, resolent el primer problema que s'ha trobat. Per la longitud es fa servir `atan2` que rep dos arguments i pot determinar en quin quadrant es troba el punt, ajustant el resultat per retornar el valor correcte entre $-\pi$ i π . És una funció que l'arctangent simple no és capaç de realitzar. Aleshores s'implementaran al codi de nou:

```
def latitud_calcular(x, y, z):
    return numpy.degrees(numpy.arctan2(z, numpy.sqrt(x**2 + y**2)))
def longitud_calcular(x, y):
    return numpy.degrees(numpy.arctan2(y, x))
```

Per introduir les operacions s'han utilitzat funcions de Numpy que es poden trobar a la seva pàgina oficial. Llavors, es retornen els valors finals a les variables respectives latitud i longitud per ser utilitzats en el bucle que s'ha creat posteriorment.

6.3.4 Llista de latituds, longituds i creació de línia de la trajectòria

Aleshores es van crear dues llistes de longituds i latituds que contenen "[]" que significa que estan buides. S'aniran omplint més tard, a mesura que es vagin llegint les dades del fitxer. Després, es crea una línia de color blau també tota buida, esperant a ser actualitzada. El

⁴ Fórmules extretes de: Tonfilm. (2015, 6 d'octubre). *Polar, spherical and geographic coordinates*. vvvv. <https://vvvv.org/blog/2015/polar-spherical-and-geographic/>

contingut s'utilitzarà per representar la ubicació del satèl·lit en un moment concret, s'anirà actualitzant a cada iteració del bucle i com a resultat, la trajectòria del satèl·lit es forjarà.

```
llista_latituds = []
llista_longituds = []
linia, = gràfic.plot([], [], "b-")
```

6.3.5 Lectura de dades i actualització del gràfic

S'inicialitza la variable `longitud_abans` i l'equivalem a `None`, indica l'última longitud enregistrada i serà utilitzada per detectar salts grans en aquest eix.

```
fitxer = open("txtFiles/dadesGPS.txt", "r")
longitud_abans = None
```

S'obre el fitxer `dadesGPS.txt` en mode lectura i amb un bucle "for" es fa que es llegeixi línia per línia i separar els seus components per classificar-los a la variable corresponent. `x`, `y` i `z` corresponen al novè, desè, onzè element respectivament. Pot semblar estrany el fet que tot i que entre claudàtors posi 10, sigui el novè element. Això és perquè es comença a comptar des del 0 inclòs, és a dir, que el primer element tindrà la posició 0. Es converteixen en `float`, un format de dades que permet representar els nombres positius i negatius amb decimals, amb nombres reals perquè s'hauran de fer servir en les equacions `latitud_calcular` i `longitud_calcular` que retornaran la latitud i la longitud. L'hora correspon a la posició 1, el segon element, i la visibilitat és el sisè. Aquest últim és necessari que sigui convertit en format `string`, una cadena de text, per futures comparacions.

```
elements = row.split()
x = float(elements[10])
y = float(elements[11])
z = float(elements[12])
visibilitat = str(elements[1])
latitud = latitud_calcular(x, y, z)
longitud = longitud_calcular(x, y)
```

6.3.6 Detecció de canvis bruscos

Es comprova si hi ha un canvi brusc, superior a 180°, la meitat de la volta a la Terra en horitzontal, en la longitud entre dos punts consecutius. Pot passar que el satèl·lit es traslladi d'un costat a un altre del mapa i sembli que s'hagi desplaçat molt de cop, però no és veritat. Com que estem treballant en un mapa 2D, es tradueix d'aquesta manera i ocasiona una línia que creua tot el mapa, però a la vida real, en ser rodona la Terra, només es tracta d'un petit

desplaçament continu com qualsevol altre que veiem en altres punts. Per evitar aquests traços, s'ha escrit un codi amb la instrucció `if` que si l'última longitud no és `None` i la diferència amb la longitud actual és major a 180 s'esborra el contingut de les llistes. Després amb `.append()` s'afegeixen les latituds i longitud actuals a les llistes i s'actualitza `longitud_abans` per assegurar que en la següent comparació es faci amb l'últim valor i no amb un de més antic.

```
if longitud_abans is not None and abs(longitud - longitud_abans) > 180:
    llista_latituds = []
    llista_longituds = []
    llista_latituds.append(latitud)
    llista_longituds.append(longitud)
    longitud_abans = longitud
```

S'actualitzen les coordenades de la línia que representa la trajectòria, fent que mostri tots els punts acumulats fins al moment.

```
linia.set data(llista longituds, llista latituds)
```

Amb la finalitat de fer-ho més visual s'ha volgut fer ús del color vermell per indicar quan és visible la nau. L'if comprova si el satèl·lit és visible en aquell instant. Quan la variable `visibilitat` és igual a 1, ho serà i com a resultat la ruta i el punt dibuixats canviaran de color. En cas contrari, es mantindran blaus.

```
if visibilitat=='1':
    satel·lit, = ax.plot(longitud, latitud, 'ro')
    linia.set_color('red')
else:
    satel·lit, = gràfic.plot(longitud, latitud, 'bo')
```

6.3.7 Pintat de la trajectòria

Tot seguit, es força el repintat del gràfic amb els canvis recents, la nova ubicació del satèl·lit representada amb un punt i la distància de més que ha recorregut en comparació amb el dibuix anterior, amb `plt.draw()` de Matplotlib. `plt.pause()` crea una petita pausa perquè l'usuari pugui visualitzar les noves modificacions i després del temps indicat s'esborra la representació de la nau del gràfic i torna a començar el bucle.

S'anirà repetint aquest procés per cada una de les línies que contingui el fitxer `dadesGPS.txt`. Obert en mode lectura, quan acabi es tancarà.

[illegible]

```
satel·lit.remove()
plt.draw()

fitxer.close()
```

6.3.8 Avaluació

Malgrat que es va aconseguir l'objectiu general, el resultat personalment no va acabar de convèncer. Acomplia la funció, però el codi en estar lligat a un bucle tardava molt a dibuixar la ruta d'un dia d'un sol satèl·lit. En aquest cas s'estaven simulant 31 dies per cadascun, la qual cosa feia que l'execució fos molt lenta i poc eficient. Per arreglar-ho es va decidir fer servir `FuncAnimation`, una subllibreria de Matplotlib que la convertiria en una animació, fet que permetria fer salts en el temps per avançar més de pressa i ajustar l'interval temporal de manera senzilla.

En la primera versió també es van observar alguns problemes que s'intentaran resoldre en la segona. Per exemple, quan és visible converteix tot el gràfic de color vermell. Això no interessa perquè no aporta cap utilitat. Seria millor que només mostrés el tros que ho és d'aquest color i que la resta ho mantingués blau. A més a més, tan bon punt passa d'un extrem a l'altre, s'esborra el traçat dibuixat. Això pot ser bo per no tenir un mapa sobrecarregat, però s'ha pensat que seria interessant veure la trajectòria completa d'un dia abans d'esborrar-la i continuar perquè així ajudaria a entendre el seu moviment i el recorregut que ha fet. Per facilitar la interacció de l'usuari es podria crear una etiqueta al costat que mostrés informació com la posició, longitud, latitud i altres dades sobre el punt actual del satèl·lit.

6.4 Segona versió

Malgrat el que s'ha comentat anteriorment, no significa que la primera versió no tingués utilitat. Per la nova es van fer servir moltes de les mecàniques que s'havien implementat, com ara per exemple l'ús de les llistes buides que s'actualitzen a mesura que es processen les dades, així com la funció de transformació de les coordenades ECEF a latitud i longitud. Parts com importacions i configuracions de mapes també es van poder reutilitzar.

6.4.1 Importació de les llibreries i configuració del mapa

Es van mantenir gairebé les mateixes línies de codi que la primera versió. En el cas de les importacions se n'han afegit dues més. La primera és per raó de comoditat, del submòdul de Matplotlib: `matplotlib.widgets` s'ha carregat `Button`, un element interactiu que permet a l'usuari interactuar amb el gràfic a través d'un botó, realitzant això s'hi podrà accedir de manera més directa. No caldrà escriure `plt.widgets.Button()` cada vegada, amb un

simple `Button()` ja es podrà crear. Ha resultat molt útil tenint en compte que se n'han implementat vuit. La segona és perquè `animation` no està exposat directament dins del mòdul principal, significa que si s'intenta cridar `matplotlib.animation.FuncAnimation()` no funcionarà. Per aquesta raó, cal importar-lo explícitament del submòdul `Matplotlib.animation`.

```
from matplotlib.animation import FuncAnimation
from matplotlib.widgets import Button
```

Pel que fa a la configuració del mapa, només s'ha afegit la línia final `gràfic.set_position()`. Mou el mapa cap a l'esquerra per poder deixar un espai al costat dret per poder-hi col·locar els botons de control i la informació sobre la posició del satèl·lit.

```
gràfic.set_position([0.06, 0.02, 0.75, 0.85])
```

6.4.2 Funció transformació de coordenades ECEF a latitud i longitud

S'ha conservat igual que el primer codi, ja que complia correctament la seva finalitat i en aquest segon també s'havia de treballar amb latitud i longitud.

6.4.3 Processament de dades de posició

Actualment, totes les diferents dades de cada satèl·lit en instants distints estan junts en un mateix document `.txt` que s'ha denominat amb la variable `fitxer`. Dintre del nou codi es crearà una llista de dades per facilitar l'accés després. Així no haurem de llegir el fitxer repetidament. Es comença obrint-lo en mode lectura i després cal reescriure el contingut dins de la llista buida anomenada `dades`.

```
fitxer=open("txtFiles/dadesGPS.txt", "r")
dades = []
```

S'omplirà amb el bucle:

```
for fila in fitxer:
    elements = fila.split()
    ID_actual = elements[0]
    x = float(elements[10])
    y = float(elements[11])
    z = float(elements[12])
    latitud = latitud_calcular(x, y, z)
    longitud = longitud_calcular(x, y)
    visibilitat = elements[1]
```

```

hora = elements[5]
dia = str(elements[2]) + " " + str(elements[3]) + " " + str(elements[4])
dades.append((longitud, latitud, visibilitat, hora, dia, ID_actual))

```

Es dedica a separar els components de cadascuna de les files del document i assignar-los a la variable corresponent. El primer element de la línia, [0], perquè es comença a comptar des d'aquesta xifra, correspon a l'identificador que es guardarà a `ID_actual`, el desè element com a `x`, l'onzè com a `y` i el dotzè com a `z`. És necessari convertir aquests en específics en format `float` perquè s'hauran d'utilitzar per a operacions a `latitud_calcular(x, y, z)` i `longitud_calcular(x,y)`. Aquest tipus de dada permet representar els nombres positius i negatius amb decimals, amb nombres reals. La visibilitat correspon al segon element, l'hora el sisè i la data està format pel tercer, quart i cinquè. `.append()` ho afegeix tot en una mateixa línia en la llista `dades`.

6.4.4 Línies de la trajectòria

Es comença usant `.plot()`, que crea línies i punts buits indicats amb `[]` que s'aniran actualitzant amb les coordenades del satèl·lit a posteriori. La trajectòria quan la nau és invisible es vol que sigui de color blau, en cas contrari serà de color vermell.

```

linia, = gràfic.plot([], [], color="blue")
linia_vermella, = gràfic.plot([], [], color="red")
punt_actual, = gràfic.plot([], [], "o")

```

Llavors, també se'n creen per les longituds i latituds de la línia blava i de la vermella que s'emplenaran després.

```

llista_longituds = []
llista_latituds = []
llista_longituds_vermell = []
llista_latituds_vermell = []

```

Adicionalment, també es preparen les variables que es faran servir més tard.

```

longitud_abans = None

```

Indicarà la longitud anterior a la que es dibuixa, posem `None` a l'inici per transmetre que encara no s'ha emmagatzemat cap valor.

```

ID_abans = None

```

Servirà per recordar l'identificador del satèl·lit anterior per comparar-lo amb l'actual.

```

dia_anterior=None

```

Té la mateixa finalitat que `ID_abans`, però guarda la data del dia anterior.

```
fotograma=[0]
```

És una llista amb el 0 com a sol element que anirà essent substituït durant el programa per comptar quants fotogrames s'han processat.

6.4.5 Text

Per crear la llegenda que mostrarà informació sobre el punt actual es fa servir `.text()`. El primer element dintre els parèntesis s'hi col·locarà la posició concreta que es vol que estigui situat el text dintre de la figura; en el segon hi haurà el contingut, inicialment també estaran buits perquè es completin més tard i després s'escriuran les altres propietats com la mida de la lletra, color i estil.

```
text_info = figura.text(0.85, 0.9, "Informació", fontweight="bold", fontsize=11)
text_visibilitat = figura.text(0.85, 0.66, "", fontsize=10, color="black")
text_hora = figura.text(0.85, 0.72, "", fontsize=10, color="black")
text_posicio = figura.text(0.85, 0.84, "", fontsize=10, color="black")
text_dia = figura.text(0.85, 0.78, "", fontsize=10, color="black")
```

Es titula el gràfic “Trajectòria dels satèl·lits:” que es mostrarà a la part superior. A mesura que avanci el codi, al costat hi apareixerà l'identificador del qual s'està visualitzant en la mida de lletra 14.

```
gràfic.set_title(f"Trajectòria dels satèl·lits: {ID_actual}", fontsize=14)
```

6.4.6 Funció “actualitzar”

La funció que s'encarregarà d'anar renovant la trajectòria s'anomenarà `actualitzar`.

En primer lloc, utilitzant `global` s'han universalitzat les variables que s'utilitzaran en aquest procés perquè no només estiguin disponibles en la respectiva funció local.

```
def actualitzar(f):
    global longitud_abans, ID_abans, llista_longituds, llista_latituds,
    dia_anterior, llista_longituds_vermell, llista_latituds_vermell
```

Després s'ha agafat la llista `dades`. `f` indica la posició de la línia la qual els seus elements seran repartits en les variables:

```
longitud, latitud, visibilitat, hora, dia_actual, ID_actual = dades[f]
```

El primer element de la fila amb l'índex marcat per la xifra de `frame` es tresorejarà a `longitud`, el segon a `latitud`, el tercer a `visibilitat`, el quart a `hora`, el cinquè a `dia_actual` i el sisè a `ID_actual`.

Llavors s'han escrit les condicions. D'entrada es vol que quan es representi la trajectòria d'un satèl·lit diferent de l'anterior es reinici la trajectòria. No es desitja que es continuï dibuixant de tal manera que apareguin línies contínues unint punts de naus diferents, ja que semblaria que totes formen part d'un mateix trajecte. Emprant la instrucció `if`, si `ID_abans` no és igual a `None` i és diferent a `ID_actual`, en altres paraules que presenta un canvi, s'esborraran els continguts de les llistes i es restabliran `longitud_abans` i `dia_anterior`. A més, es canviarà el títol per "Trajectòria del satèl·lit:" i l'identificador de la nova nau.

```
if ID_abans != None and ID_actual != ID_abans:
    llista_longituds = []
    llista_latituds = []
    llista_longituds_vermell = []
    llista_latituds_vermell = []
    longitud_abans = None
    dia_anterior = None
    gràfic.set_title(f"Trajectòria del satèl·lit: {ID_actual}", fontsize=14)
```

També es vol que es reiniciï quan es passa d'un dia a un altre. Cal recalcar que els satèl·lits en òrbita baixa fan una mitjana de 16 voltes a la Terra per dia, si no es fa, en visualitzar la ruta de 31 dies quedaria tot el mapa pintat, com a resultat seria impossible veure res. Es repeteix la mateixa mecànica, si `dia_anterior` no és `None` i és diferent que `dia_actual` es netegen les llistes i es restableix `longitud_abans`.

```
if dia_anterior != None and dia_actual != dia_anterior:
    llista_longituds = []
    llista_latituds = []
    llista_latituds_vermell = []
    llista_longituds_vermell = []
    longitud_abans = None
```

Igual que la primera versió per evitar que quan el satèl·lit passi d'un extrem a l'altre de manera sobtada i dibuixi una línia que travessi tot el mapa s'ha d'afegir una condició. Quan la `longitud_abans` no és `None` i el valor absolut de la diferència entre `longitud` (`longitud actual`) i `longitud_abans` supera els 180° s'insereix el valor especial `NaN` (not a number) que forma part de la llibreria `Numpy` i indica la manca d'una xifra. Fa que el segment del tram sigui invisible, així trencant la trajectòria en aquest punt.

```
if longitud_abans != None and abs(longitud - longitud_abans) > 180:
    llista_longituds.append(numpy.nan)
    llista_latituds.append(numpy.nan)
```

S'han guardat la longitud i latitud a les seves llistes corresponents. Es vol que es torni a representar les dades una vegada realitzat el salt.

```
llista_longituds.append(longitud)
llista_latituds.append(latitud)
```

S'han actualitzat les variables que descriuen dades anteriors. Això provoca que quan la funció es torni a cridar contingui el punt que acabes de processar, que són les anteriors per la consegüent línia. Si no s'hagués realitzat, aquí s'estaria comparant un valor més antic que el darrer.

```
longitud_abans = longitud
ID_abans = ID_actual
dia_anterior = dia_actual
```

Amb la finalitat d'obtenir un resultat més visual s'han marcat els trossos visibles amb el color vermell. Per aconseguir-ho s'efectua que si el valor de visibilitat és igual a 1 s'afegiran els valors de longitud i latitud a `llista_longituds_vermell` i `llista_latituds_vermell`. En cas contrari, s'afegirà el valor NaN.

```
if visibilitat == "1":
    llista_longituds_vermell.append(longitud)
    llista_latituds_vermell.append(latitud)
else:
    llista_longituds_vermell.append(numpy.nan)
    llista_latituds_vermell.append(numpy.nan)
```

Aleshores s'actualitza el que es veu al mapa després de processar les dades d'un fotograma mitjançant `.set_data()`. S'assignen a la línia les coordenades acumulades fins al moment. Conté tots els punts de la trajectòria completa. D'altra banda, `linia_vermella` només en casos que sigui visible. Quan no ho és, el valor que guarda és NaN, això visualment s'aprecia com un tall a la línia vermella contínua, ocasionant que es converteixi en un segment.

```
linia.set_data(llista_longituds, llista_latituds)
linia_vermella.set_data(llista_longituds_vermell, llista_latituds_vermell)
```

La posició actual que és la més recent dintre de les llistes es marca amb un punt i s'anirà actualitzant a mesura que avanci l'animació.

```
punt_actual.set_data([longitud], [latitud])
```

Dibuixa totes les coordenades guardades en les llistes. Si el satèl·lit no és visible en una posició, la línia de la trajectòria serà blava, però si ho és, el tram en què és observable des de l'institut Pere Alsius serà de color vermell. Això és degut que quan és invisible s'afegeix el valor NaN que fa que no sigui apreciable.

Per fer que el punt es torni vermell s'ha fet servir la mateixa estructura que per `linia`, utilitzant la instrucció `if`, `else` i la funció `.set_color()`. Pel que fa a la taula d'informació que es va actualitzant per cada instant, s'aprofitarà per introduir la part que depèn de la visibilitat a la condició. `.set_text()` s'encarregarà d'actualitzar el contingut del text. En aquest cas, s'ha fet que es mostri una pregunta: "Visible?" i depenent del valor de visibilitat s'escriurà "Sí" o "No".

```
if visibilitat=="1":
    punt_actual.set_color("red")
    text_visibilitat.set_text("Visible?: Sí")
else:
    punt_actual.set_color("blue")
    text_visibilitat.set_text("Visible?: No")
```

Per altres magnituds, es fa que segueixi la següent estructura: "el seu nom: el valor corresponent". S'escriu `f` davant per convertir-ho en format `f-string` que permet redactar el contingut d'una variable utilitzant `{}`. En el cas de `text_posicio`, s'ha d'arrodonir la `longitud` i la `latitud` als dos decimals per evitar mostrar moltes xifres.

```
text_hora.set_text(f"Hora: {hora}")
text_posicio.set_text(f"Posició: ({latitud:.2f}, {longitud:.2f})")
text_dia.set_text(f"Dia: {dia_actual}")
```

6.4.7 Botons de control

Amb l'objectiu de facilitar la navegació temporal s'han creat botons que permeten fer salts directes en el temps sense haver d'esperar que la representació avanci en temps real.

6.4.7.1 Salts endavant

S'han començat definint les funcions. Es redomina el `fotograma[0]` que conté el fotograma corresponent al moment actual, com a `frame_actual`. El nombre que se saltaran s'aconsegueix multiplicant 60 (la quantitat de fotogrames per hora) pel nombre d'hores diàries per obtenir els fotogrames per dia. Si es vol obtenir la quantitat per satèl·lit, es fa el

producte d'entre el resultat d'abans i els dies totals que se simulen. El resultat del desplaçament que es desitja realitzar es guarda a la variable `salt`.

Per passar a la següent hora, `salt` seria igual a 60.

A fi de saltar de dia es faria $\frac{60 \text{ fotogrames}}{1 \text{ hora}} \times \frac{24 \text{ hores}}{1 \text{ dia}}$ i la variable equivaldria al resultat, 1440.

I pel canvi de satèl·lit $\frac{60 \text{ fotogrames}}{1 \text{ hora}} \times \frac{24 \text{ hores}}{1 \text{ dia}} \times \frac{31 \text{ dies}}{1 \text{ satèl·lit}}$, resultaria 44640.

Un cop avançat en el temps, es posiciona en el fotograma `nou_frame`, obtingut sumant el valor de `salt` al fotograma original en què es trobava. Aleshores es fa ús d'un bucle "for" i es crea un rang de valors que va des del que ve immediatament posterior a l'actual (`frame_actual+1`) fins al que volem avançar, incloent-lo (`nou_frame+1`). Per cada valor dins de l'interval es crida la funció que s'ha definit abans `actualitzar()` que va actualitzant el gràfic. Després, el contingut de `nou_frame` s'introdueix a `fotograma` perquè a partir d'aquell instant es faci servir el nou fotograma al qual hem saltat per executar els altres fragments de codi de l'animació. `actualitzar()` no inclou la funció de mostrar el nou gràfic amb els canvis realitzats a l'usuari, és per aquesta raó que es força a redibuixar-lo amb la instrucció `fig.canvas.draw_idle()`. Això fa que la trajectòria quedi directament dibuixada fins al nou punt i no es mostri cada pas.

```
def saltar_dia(d):
    frame_actual = fotograma[0]
    salt = 1440
    nou_frame = frame_actual + salt
    for f in range(frame_actual + 1, nou_frame + 1):
        actualitzar(f)
    fotograma[0] = nou_frame
    fig.canvas.draw_idle()
```

Per `saltar_dia(d)` i `següent_satèl·lit(sat)` es manté tot igual menys el contingut de `salt` que es canviarà pel valor corresponent.

6.4.7.2 Pausar i avançar

Es crea la variable global `pausa` i s'estableix com a `False`, que voldrà dir que l'animació està en execució. La funció que s'encarregarà d'aturar l'animació és `pausar(_)`. Es globalitza `pausa` perquè sigui possible treballar amb ella dintre del bloc i llavors se li assigna `True`. Després se'n configura un altre, `continuar(_)` que fa el contrari, el transforma de nou a `False`.

```
pausa=False
```

```
def pausar(_):
    global pausa
    pausa=True
```

```
def continuar(_):
    global pausa
    pausa=False
```

A continuació, es crea `avança(_)` que s'executarà cada cop que `FuncAnimation` demani un nou pas per renovar l'animació. Controla quin fotograma mostrar. Quan `pausa` és `True`, no fa res i es manté al mateix fotograma, si no s'incrementa perquè l'animació procedeixi i s'aplica `actualitzar(fotograma[0])` perquè la línia, el punt i els textos canviïn a les noves dades.

```
def avança(_):
    if pausa==True:
        fotograma[0]=fotograma[0]
    else:
        fotograma[0] += 1
        actualitzar(fotograma[0])
```

6.4.7.3 Salt enrere

La funció `reconstruir_fins_frame` serà la clau per tots els salts temporals cap enrere que es faran. El seu objectiu és reconstruir l'estat del mapa fins al fotograma que se salta, així dona l'efecte que ha esborrat els que venien després. Rep dos arguments `frame_actual`, el número del fotograma al qual està i `salt` que indica quant volem retrocedir. Primer de tot, es reinicien les llistes on es guarden la latitud i la longitud i les variables `longitud_abans`, `ID_abans`, `dia_anterior` perquè estem començant de zero.

Per saltar al satèl·lit anterior es redibuixa la trajectòria des del primer fotograma de la nau precedent fins al qual es vol arribar. Al principi s'havia executat sense aquest fragment i s'ha vist que és necessari perquè no és possible dibuixar una gran quantitat de fotogrames a la vegada, ja que pot ocasionar trajectòries errònies i no contínues. Es calcula la seva posició dins de la llista de satèl·lits, dividint de manera entera amb `"/"` el `nou_frame` entre el `salt`. Llavors el resultat es multiplica per `salt` per aconseguir el primer fotograma del satèl·lit anterior al que s'està. Aleshores es recorren tots els elements entre el `primer_frame` i `nou_frame`, incloent-hi aquest últim. El codi que ve després es reutilitza d'`actualitzar()`

perquè cal aplicar la mateixa lògica de construcció de la trajectòria, s'afegeixen punts a les llistes de longitud i latitud, es gestionen els grans salts de longitud, es comproven els canvis de dia o de satèl·lit i s'assignen els punts visibles en vermell.

Pel que fa a altres tipus de retrocés temporal es manté tot igual que pels satèl·lits amb l'excepció del bucle, es recorre des del fotograma 0 fins al `frame_nou`, inclòs.

Ara es defineixen les funcions que s'encarregaran de dibuixar els salts. Com en el cas dels salts temporals cap endavant s'inicia redenant el `fotograma[0]` com a `frame_actual`. Es defineix el salt corresponent i per obtenir el `nou_frame` a diferència de l'altre procés, es resta `salt` a `frame_actual`. S'introdueix `fotograma[0]` a `nou_frame` i es criden les funcions `reconstruir_fins_frame(nou_frame, salt)` i `actualitzar(nou_frame)`, el primer s'encarrega de recalculer l'estat del mapa i el darrer de fer el dibuix efectiu.

6.4.7.4 Connectar funcions als botons

Es creen els espais dins de la figura on es veurà el botó mitjançant `plt.axes([marge esquerre, marge inferior, amplada, alçada])`. Van a un rang de valor del 0 a 1. Actuen com un percentatge de la mida total de la figura, de manera que en són relatius. Per exemple pel botó continuar, s'ha col·locat al punt que és el 90% de l'amplada total, al 20% d'alçada i per les seves mides la seva amplada és el 8% de la total de la figura i l'alçada un 5%.

```
mides_continuar = plt.axes([0.90, 0.2, 0.05, 0.05])
```

Els valors provenen d'una estimació aproximada i posteriorment s'han acabat d'ajustar amb proves successives fins a trobar la ubicació adequada. Se'ls ha donat a totes la mateixa dimensió, amb l'excepció de `pausa` i `continuar` que se'ls hi ha reduït el tamany per poder-se col·locar de costat. És per això que tots tenen gairebé les mateixes coordenades, l'única que varia d'una a l'altra és el segon argument de la funció. S'ha vist que l'espai ideal entre cadascun és 0.05 de manera que s'ha partit de la xifra de `ax_boto_anterior_sat` i se li ha anat restant aquest valor. Com a resultat, s'ha pogut garantir una disposició visual ordenada.

```
mides_boto_anterior_sat = plt.axes([0.85, 0.36, 0.1, 0.04])
mides_boto_retrocedir_d = plt.axes([0.85, 0.31, 0.1, 0.04])
mides_boto_retrocedir_h = plt.axes([0.85, 0.26, 0.1, 0.04])
mides_pausa = plt.axes([0.85, 0.2, 0.04, 0.05])
mides_continuar = plt.axes([0.90, 0.2, 0.05, 0.05])
```

```
mides_següent_hora = plt.axes([0.85, 0.15, 0.1, 0.04])
mides_dia = plt.axes([0.85, 0.10, 0.1, 0.04])
mides_següent_sat= plt.axes([0.85, 0.05, 0.1, 0.04])
```

Es creen els botons amb `Button()` que requereix l'espai on s'ubicaran i el text que es mostrarà.

```
boto_anterior_sat= Button(ax_boto_anterior_sat, 'Anterior satèl·lit')
boto_retrocedir_d = Button(ax_boto_retrocedir_dia, 'Retrocedir 1 dia')
boto_retrocedir_h = Button(ax_boto_retrocedir_hora, 'Retrocedir 1h')
boto_pausa = Button(mides_pausa, 'Pausa')
boto_continuar = Button(mides_continuar, 'Continuar')
boto_saltar_hora = Button(mides_següent_hora, 'Avançar 1h')
boto_saltar_dia = Button(mides_dia, 'Avançar 1 dia')
boto_següent_sat = Button(mides_següent_sat, 'Següent satèl·lit')
```

Després es connecten els botons amb les seves pertinents funcions que s'han definit als subapartats anteriors utilitzant `.on_clicked()`.

```
boto_anterior_sat.on_clicked(anterior_satèl·lit)
boto_retrocedir_d.on_clicked(retrocedir_dia)
boto_retrocedir_h.on_clicked(retrocedir_hora)
boto_pausa.on_clicked(pausar)
boto_continuar.on_clicked(continuar)
boto_saltar_hora.on_clicked(saltar_hora)
boto_saltar_dia.on_clicked(saltar_dia)
boto_següent_sat.on_clicked(següent_satèl·lit)
```

Pel cas de pausar i continuar, en ser més petites, s'ha hagut d'ajustar la mida de la lletra.

```
boto_continuar.label.set_fontsize(8)
boto_pausa.label.set_fontsize(8)
```

Finalment, s'ha afegit el títol "Botons de control".

```
text_botons=figura.text(0.85, 0.41, "Botons de control",
fontweight="bold", fontsize=10)
```

6.4.8 FuncAnimation i mostrar dibuix

Per últim, es crea l'animació utilitzant `FuncAnimation` de Matplotlib que rep com a arguments la figura on es mostrarà, la funció que actualitza cada fotograma i el temps en mil·lisegons entre cada actualització. D'opcionals que s'han utilitzat en aquest cas és `blit`

que determina si es torna a dibuixar tota la figura a cada fotograma o només els elements que canviïn i `repeat` que controla si l'animació es repeteix automàticament.

```
animació = FuncAnimation(figura, avança, interval=30, blit=False, repeat=False)
plt.show()
```

6.4.9 Comprovació

No obstant això, després de fer comprovar el funcionament del codi, sense voler s'ha premut `boto_anterior_sat` en el primer satèl·lit, fet que ha portat a un dibuix erroni. Per evitar casos similars s'ha establert controls que limitin les accions de l'usuari per no provocar avanços o retrocessos fora dels rangs vàlids de dades.

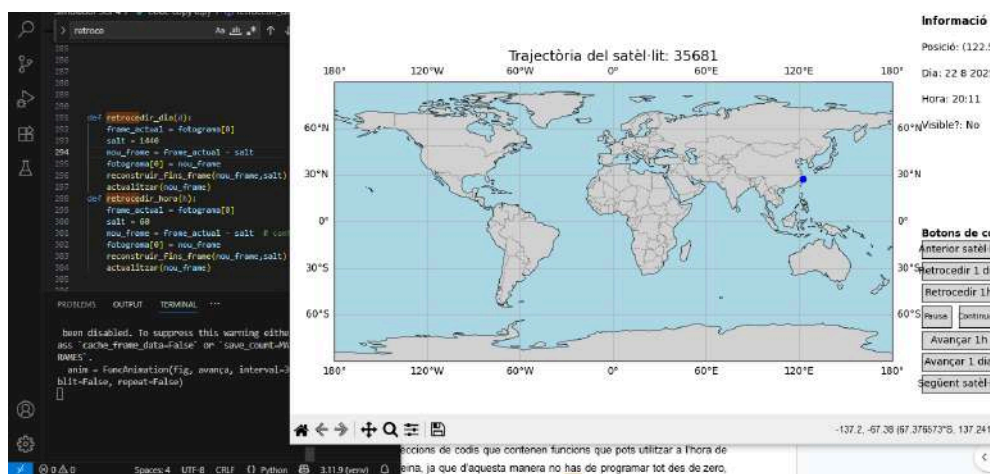


Fig. 28 Resultat erroni que prové de prémer el botó “anterior satèl·lit” en el primer satèl·lit.
Font: pròpia

Per tots els botons que permeten fer salts temporals futurs s'ha implementat una línia de codi que diu que si `nou_frame` és igual o més gran que el nombre de fotogrames totals menys `salt`, no es farà res.

```
if nou_frame >= len(dades):
    return
```

En canvi, pels que fan salts cap al passat, es mantindrà igual si `nou_frame` és igual o menor a `salt`.

```
if frame_actual <= salt:
    return
```

Per comprovar que utilitzi les dades adequades s'ha anat a la línia de dades `GPS.txt` que marcava el mateix dia i hora i s'han transformat les coordenades en latitud i longitud. S'han aplicat les fórmules de l'apartat 6.3.3.

$$latitude = \arcsin\left(\frac{5452,99}{\sqrt{(-4212,63)^2 + (1606,09)^2 + (5452,99)^2}}\right) = 50,42^\circ$$

$$longitude = \text{atan2}(1606,09, -4212,63) = 159,13^\circ$$

Line	Satellite ID	Epoch	Date	Time	Altitude (km)	Azimuth (deg)	Range (km)	Range Rate (km/s)	Position (km)
12	49260	0	25	7 2025 00:11	-30.5 (deg)	4.6 (deg)	-2633.1713974	410.6031189	6551.9417598
13	49260	0	25	7 2025 00:12	-32.3 (deg)	6.7 (deg)	-2979.8205546	651.1926750	6382.1358549
14	49260	0	25	7 2025 00:13	-34.1 (deg)	8.8 (deg)	-3312.3912187	892.1378384	6186.4992573
15	49260	0	25	7 2025 00:14	-35.9 (deg)	10.8 (deg)	-3629.5468404	1132.3409799	5965.8198312
16	49260	0	25	7 2025 00:15	-37.6 (deg)	12.8 (deg)	-3930.0218797	1370.6957092	5720.9868185
17	49260	0	25	7 2025 00:16	-39.3 (deg)	14.8 (deg)	-4212.6270031	1606.0919598	5452.9874136
18	49260	0	25	7 2025 00:17	-41.1 (deg)	16.9 (deg)	-4476.2541313	1837.4213066	5162.9027318
19	49260	0	25	7 2025 00:18	-42.8 (deg)	18.9 (deg)	-4719.8802782	2063.5815686	4851.9043750
20	49260	0	25	7 2025 00:19	-44.4 (deg)	20.9 (deg)	-4942.5727814	2283.4829942	4521.2486484

Fig. 29 Moment de la simulació seleccionada
Font: pròpia

El resultat ha coincidit amb la ubicació escrita en l'etiqueta d'informació.

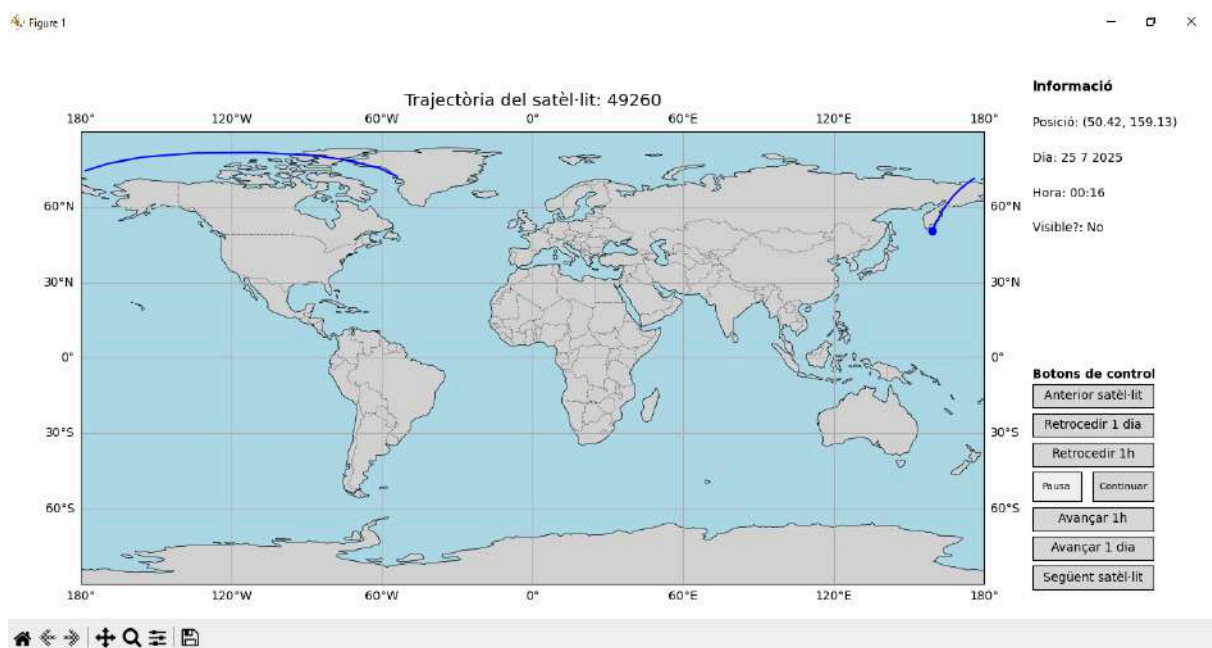


Fig. 30 Comprovació de les coordenades del mapa. Posició de LandSat-9 el 25/7/2025 a les 00:16 en el mapa de l'animació.
Font: pròpia

Per verificar que les coordenades geogràfiques estan ben configurades, s'han introduït els valors a Google Maps. Comparant els dos mapes, es veu que marquen al mateix lloc.

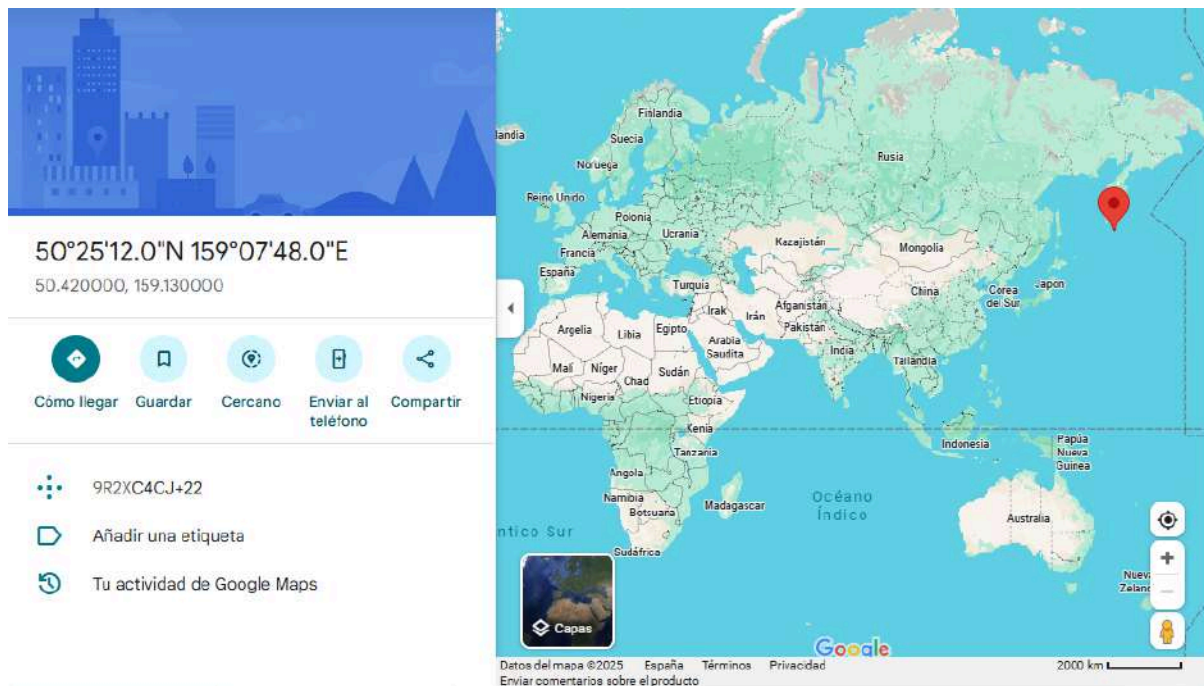


Fig. 31 Comprovació de les coordenades del mapa. Posició de LandSat-9 en Google Maps.
Font: pròpia: <https://www.google.com/maps/>

Més tard es va repetir unes quantes vegades més per assegurar que no hi hagués cap error.

7. RESULTATS I INTERPRETACIÓ DE LES TRAJECTÒRIES DIBUIXADES DURANT EL DIA 1 D'AGOST

Per respondre a la segona hipòtesi: “És possible determinar els períodes de visibilitat per cada satèl·lit per un dia concret, la quantitat de minuts totals, el moment amb la quantitat màxima i l'òptim per dur a terme una observació aficionada.” s'ha escollit el dia 1 d'agost per fer l'observació de les hores visibles dels satèl·lits i les seves trajectòries perquè es va començar a simular des del dia que es van introduir els fitxers TLE, 25 de juliol fins 1 mes més tard, el 24 d'agost i es va voler escollir un dia que estigués al mig del període. Els gràfics de visibilitat han permès aconseguir les dades de manera més visual.

7.1 LandSat-9 (49260)

Pel primer satèl·lit, LandSat s'ha vist que segueix una trajectòria que des del mapa es veu que no és recte sinó que consisteix en corbes sinusoidals que travessen quasi tota la superfície terrestre, tot i que no passen pels 90° exactes on hi ha els pols, arriben fins a molt a prop, pel que pot ser considerada una òrbita gairebé polar. La curvatura del dibuix és provocat per la projecció de la Terra, que és esfèrica, sobre la naturalesa dels mapes 2D, no vol dir que presenti un desviament. A més es veu que cada vegada que travessa la Terra longitudinalment la curvatura es desplaça cap a l'esquerra degut al gir antihorari sobre si

mateixa. Comença des del mig de l'oceà Atlàntic (47,41, -35,93) i acaba a l'Antàrtida (-65,53, 134,78). És visible durant 67 min, 1 h i 7 min, dividit en 6 períodes: 8:36-8:44 (8 min), 10:12-10:25 (13 min), 11:50-12:01 (11 min), 19:43-19:53 (10 min), 21:18-21:32 (14 min), 22:57-23:08 (11 min).

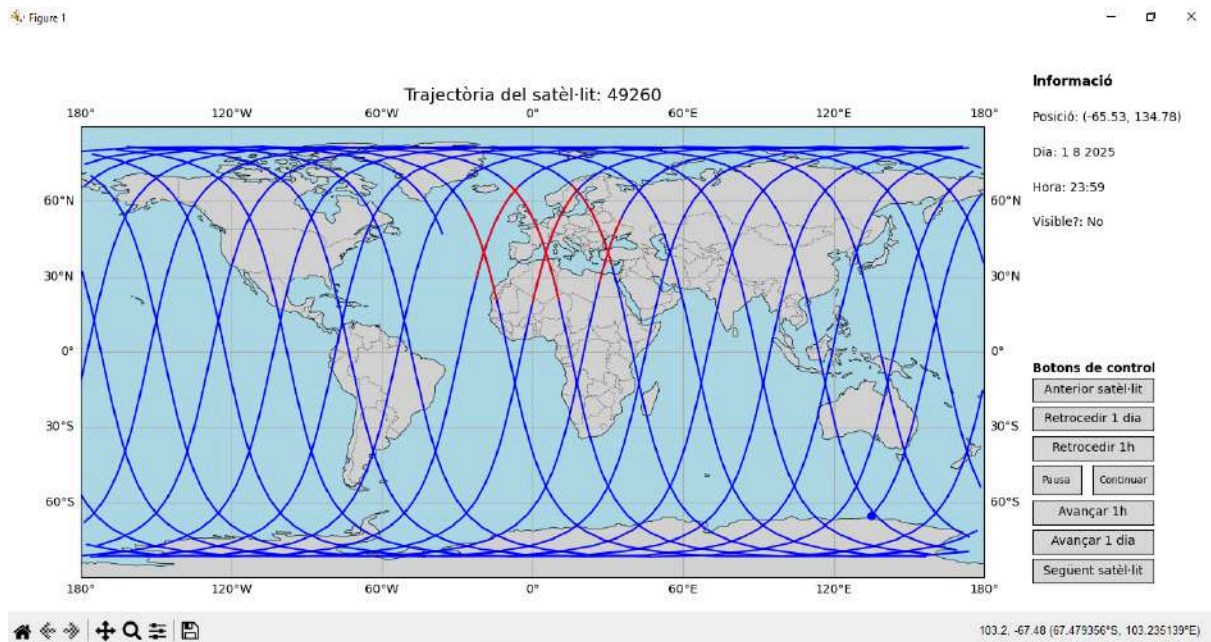


Fig. 32 Predicció de la trajectòria de LandSat-9 (49260) l'1 d'agost de 2025
Font: pròpia

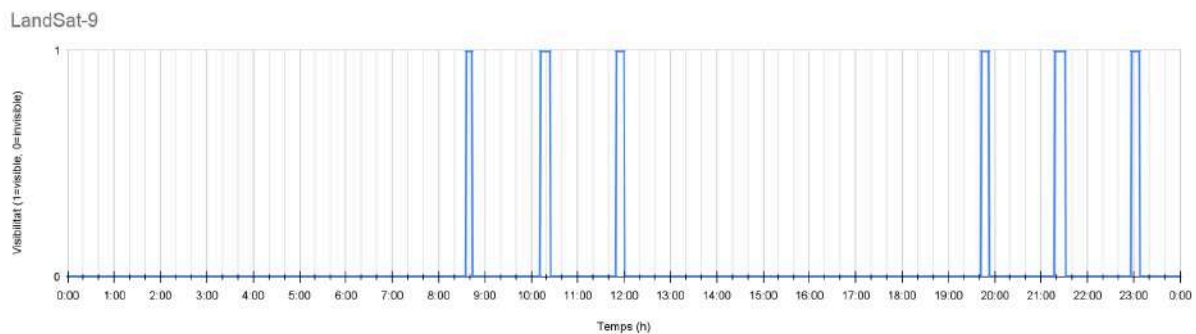


Fig. 33 Gràfica visibilitat (1=visible, 0=invisible)-temps (h) de LandSat-9 (49260) l'1 d'agost de 2025
Font: pròpia

7.2 LandSat-8 (39084)

Pel segon, LandSat-8 amb l'identificador 39084 que forma part de la mateixa constel·lació que l'anterior s'observa una trajectòria paral·lela amb les mateixes característiques. Comparteixen el mateix tipus d'òrbita, quasi polar, també disposa de 6 trams de visibilitat des de l'institut Pere Alsius i Torrent i tot i que sembli que tinguin la mateixa exacta ruta, ja que el dibuix aparenta idèntic, no és el cas. Es pot apreciar que és la trajectòria anterior, però desplaçada cap a la dreta. Comença des de just a baix d'Austràlia, a l'oceà Índic

(-47.01,144.01) i acaba a Groenlàndia (65,46, -45,15). És visible 9:23-9:36 (13 min), 11:00-11:14 (14 min), 12:40-12:47 (17 min), 18:59-19:01 (2 min), 20:30-20:42 (12 min), 22:07-22:20 (13 min). Un total de 71 min, 1 h i 11 min.

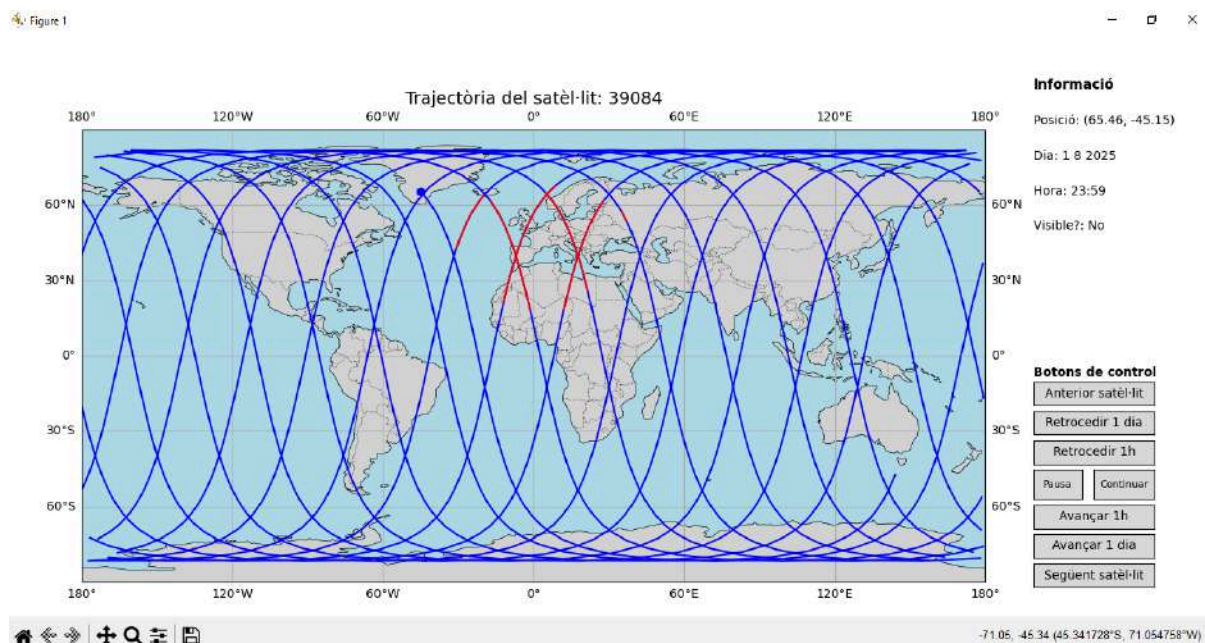


Fig. 34 Predicció de la trajectòria de LandSat-8 (39084) l'1 d'agost de 2025
Font: pròpia

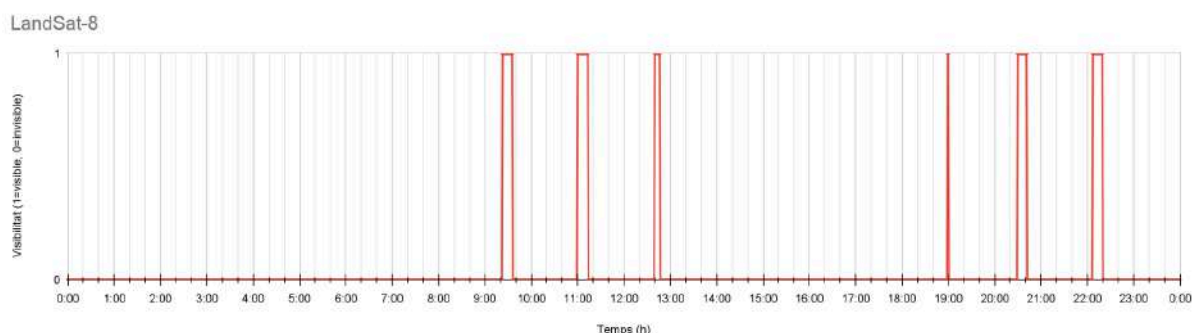


Fig. 35 Gràfica visibilitat (1=visible, 0=invisible)-temps (h) de LandSat-8 (39084) l'1 d'agost de 2025
Font: pròpia

7.3 Terra (25994)

El tercer satèl·lit és 25994 amb nom col·loquial Terra. La trajectòria és molt similar a les que s'han vist abans, però aquesta comença a l'Antàrtida al punt (-69,13, 117,94) i finalitza a l'oceà Àrtic, al nord-est de Rússia (75,18, 171,91). Es pot veure durant 67 min, 1 h i 7 min: 7:39-7:47 (8 min), 9:15-9:28 (13 min), 10:53-11:04 (11 min), 18:45-18:54 (9 min), 20:19-20:33 (14 min), 21:58-22:10 (12 min).

No obstant això, se sap que no forma part de la mateixa constel·lació que la primera i la segona perquè si en l'animació ens fixem en la seva evolució durant el pas del temps es pot

veure que aquestes primeres mantenen una diferència longitudinal constant, mentre que aquesta va variant en respecte als satèl·lits que en formen part.

Si comparem les imatges durant el dia 1 d'agost (Fig. 32, 34 i 42) s'aprecia que entre els satèl·lits de la constel·lació LandSat hi ha una diferència de 180° i que entre LandSat-9 i Terra és de 37° . Es pot concloure de manera visual, observant quants requadres hi ha a l'eix d'abscisses entre els dos. En aquest primer cas són 3 requadres, cada un equival a 60° pel que n'obtenim 180. El segon cas, però, no és possible fer-ho d'aquesta manera perquè no acaba de ser exacte, així doncs s'ha fet valor absolut de la longitud de la primera nau menys el de la segona: $|134,78 - 171,91| = 37,13^\circ$. Aquests processos s'han repetit per cada vegada que s'ha necessitat veure la diferència longitudinal.

Al cap d'un dia, el 2 d'agost a les 3:57, es percep que la longitud entre LandSat-9 i LandSat-8 continua essent 180° . Entre LandSat-9 i Terra la diferència passa a ser de 150° .

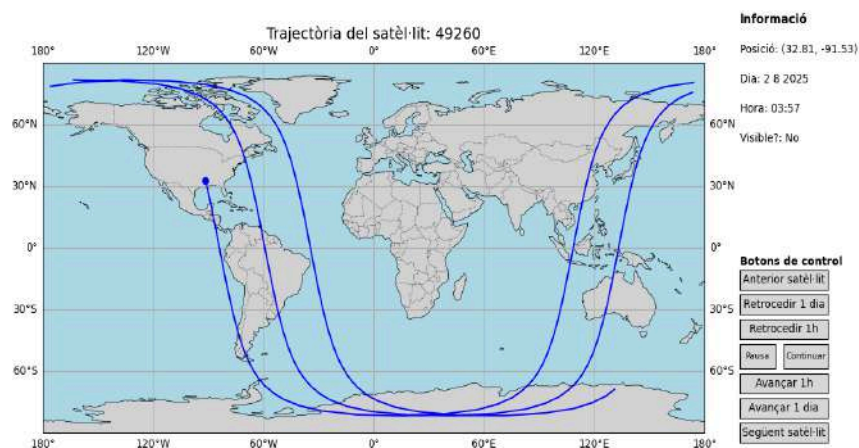


Fig. 36 Posició de LandSat-9 (49260) el 2 d'agost de 2025 a les 3:57
Font: pròpia

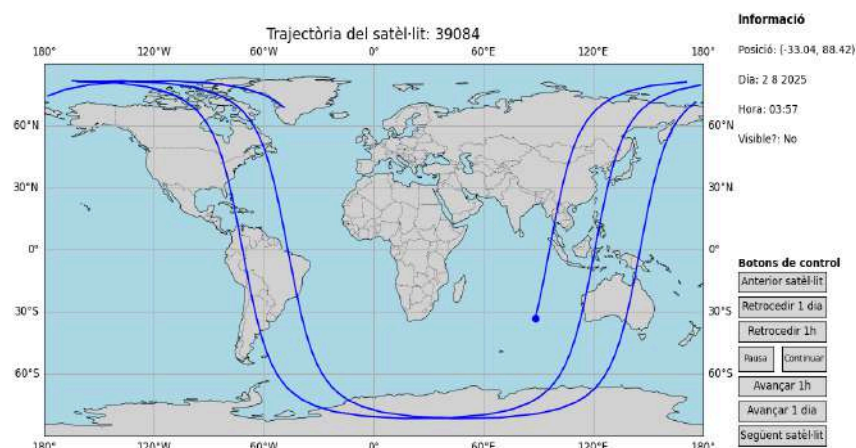


Fig. 37 Posició LandSat-8 (39084) el 2 d'agost de 2025 a les 3:57
Font: pròpia

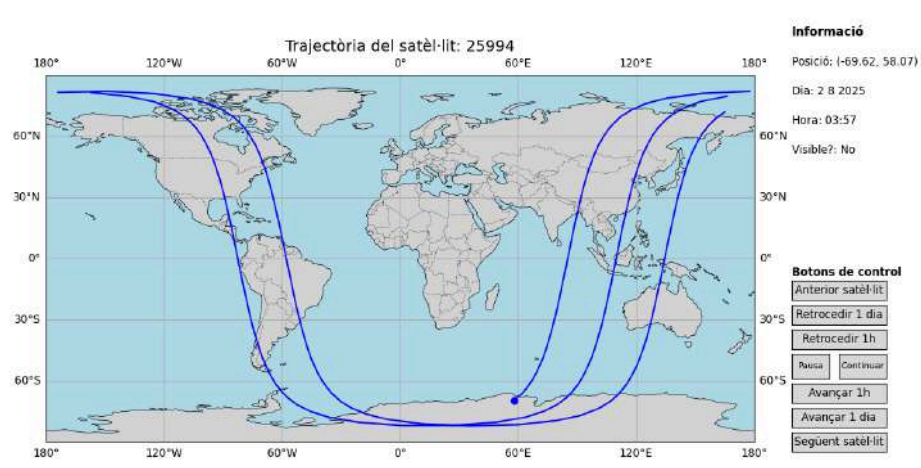


Fig. 38 Posició Terra (25994) el 2 d'agost de 2025 a les 3:57
Font: pròpia

El 9 d'agost a les 2:57 es mantenen de nou LandSat els 180°. Entre el primer i el tercer torna a variar, ara passa a ser uns 17°.

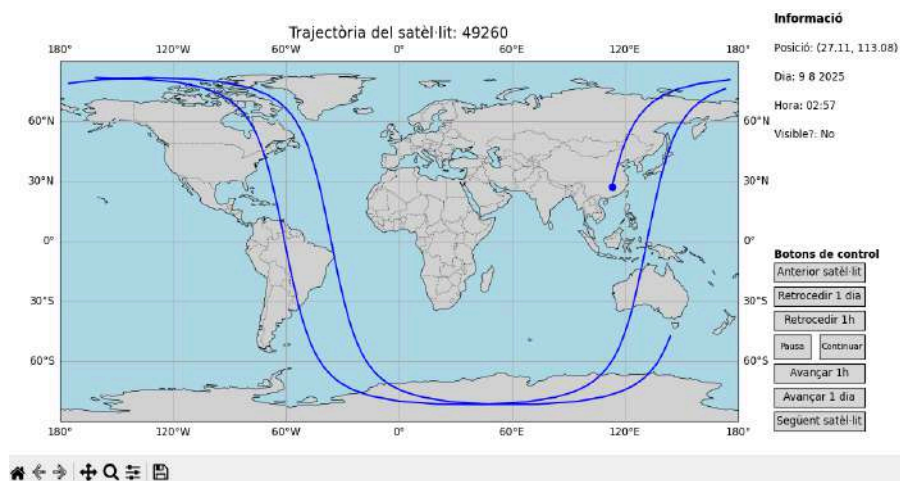


Fig. 39 Posició de LandSat-9 (49260) el 9 d'agost de 2025 a les 2:57
Font: pròpia

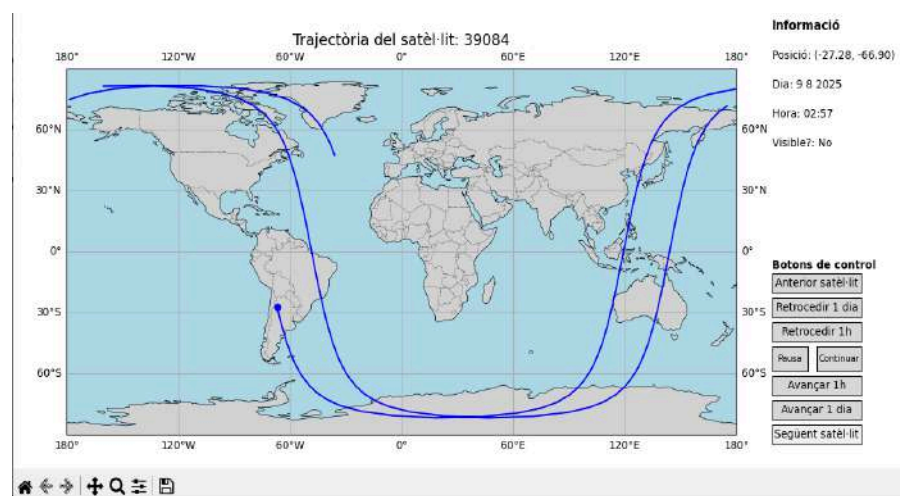


Fig. 40 Posició LandSat-8 (39084) el 9 d'agost de 2025 a les 2:57
Font: pròpia

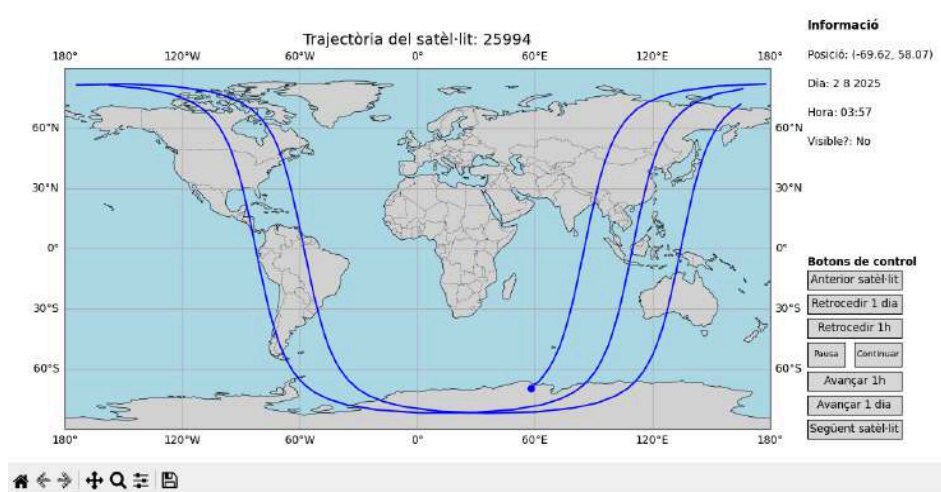


Fig. 41 Posició Terra (25994) el 9 d'agost de 2025 a les 2:57
Font: pròpia

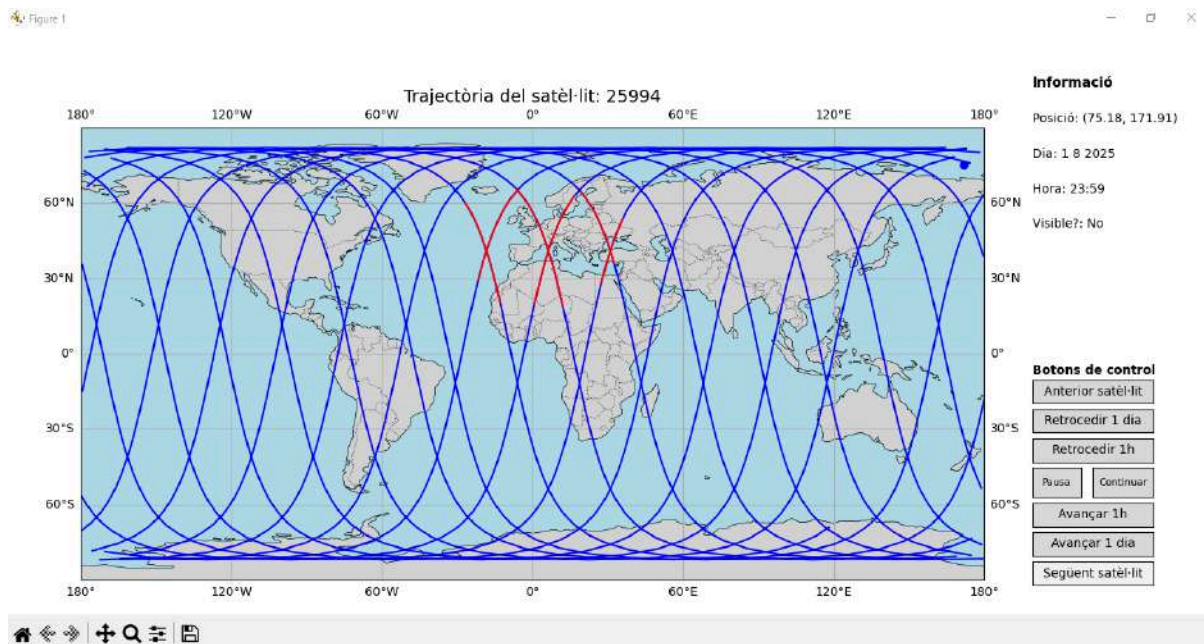


Fig. 42 Predicció de la trajectòria de Terra (25994) l'1 d'agost de 2025
Font: pròpia

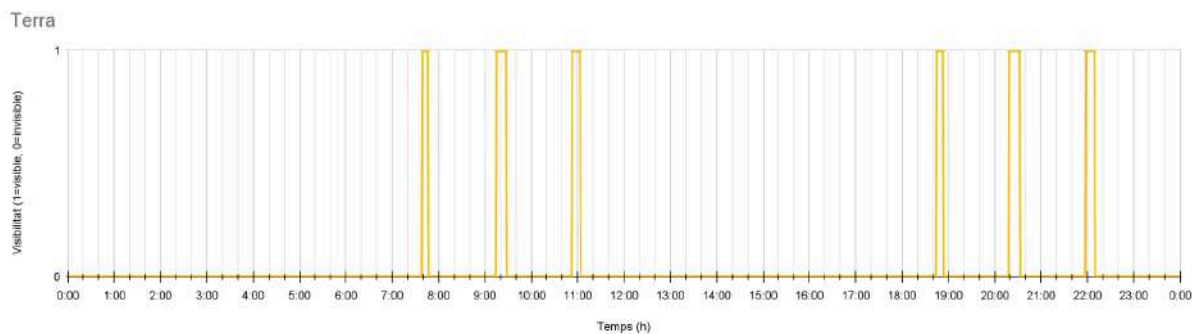


Fig. 43 Gràfica visibilitat (1=visible, 0=invisible)-temps (h) de Terra (25994) l'1 d'agost de 2025
Font: pròpia

7.4 Deimos-1 (35681) i Deimos-2 (40013)

Els dos que comparteixen el nom “Deimos” també presenten una trajectòria molt similar als anteriors. Podem concloure que tenen una òrbita quasi polar, ja que tampoc passen exactament pels pols i presenten 6 períodes de visibilitat. Deimos-1 comença a Groenlàndia (67,12, 51,37) i acaba a l'oceà Índic (-29,53, 74,70). És visible a les 1:34 (durant 1 min), 3:04-3:16 (12 min), 4:40-4:53 (13 min), 14:56-15:02 (6 min), 16:30-16:43 (13 min), 18:07-18:18 (11 min), que són un total de 56 min. Deimos-2 inicia al mig de l'oceà Atlàntic (3,36, -36,25) a finalitza a la zona sud del mateix (38,12, -30,43). Pot ser vist durant els següents períodes: 7:24-7:34 (10 min), 8:58-9:10 (12 min), 10:36-10:44 (8 min), 20:40-20:51 (11 min), 22:15-22:28 (13 min), 23:53-23:59 (6 min), que en la totalitat són 60 min, 1 hora, 4 min més que Deimos-1. S'ha pogut comprovar que no són una constel·lació, ja que la diferència longitudinal varia. Segons els resultats del 25 de juliol a les 6:10 del matí, la distància entre les dues naus és de $|(-4,62) - 54,35| = 58,97^\circ$. L'1 d'agost a les 23:59 es veu que passa a ser de $|74,70 - (-30,43)| = 105,13^\circ$.

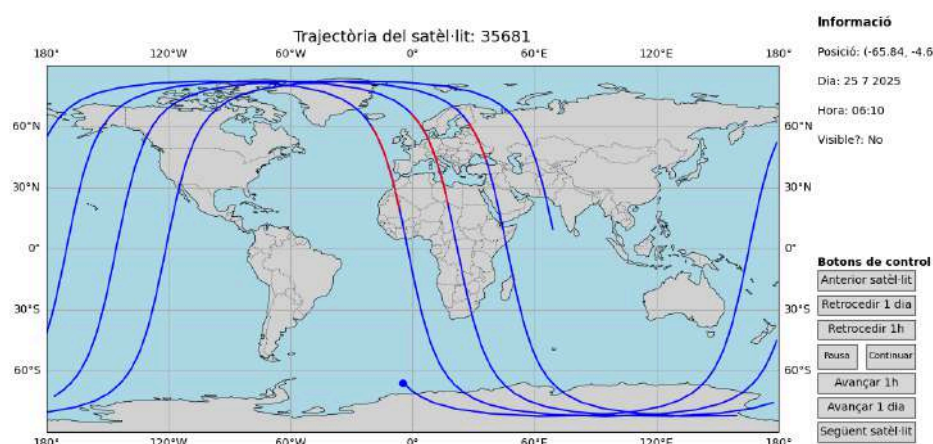


Fig. 44 Posició de Deimos-1 (35681) el 25 de juliol de 2025 a les 6:10
Font: pròpia

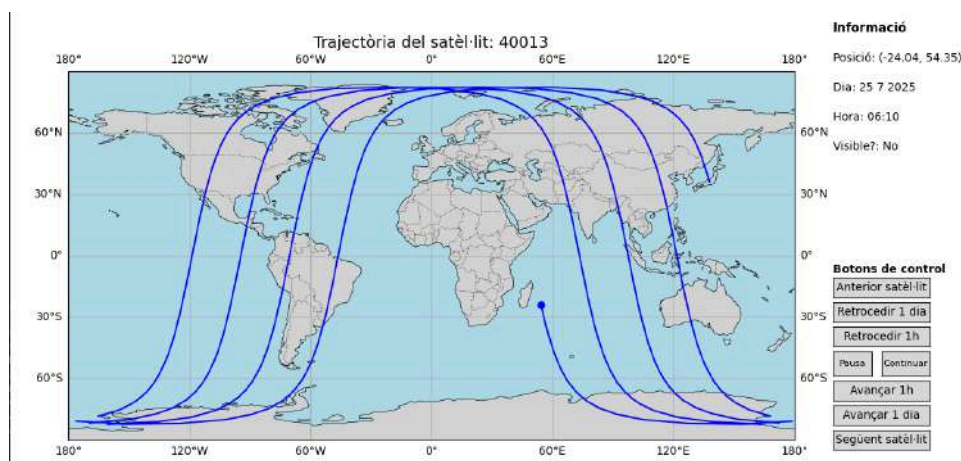


Fig. 45 Posició de Deimos-2 (40013) el 25 de juliol de 2025 a les 6:10
Font: pròpia

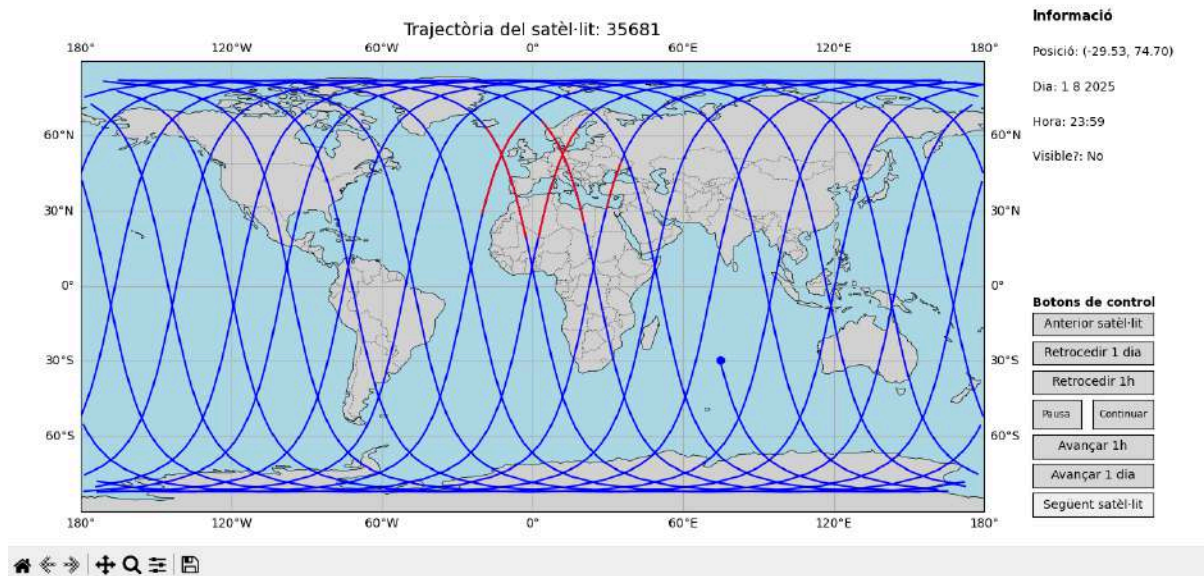


Fig. 46 Predicció de la trajectòria de Deimos-1 (35681) l'1 d'agost de 2025
Font: pròpia

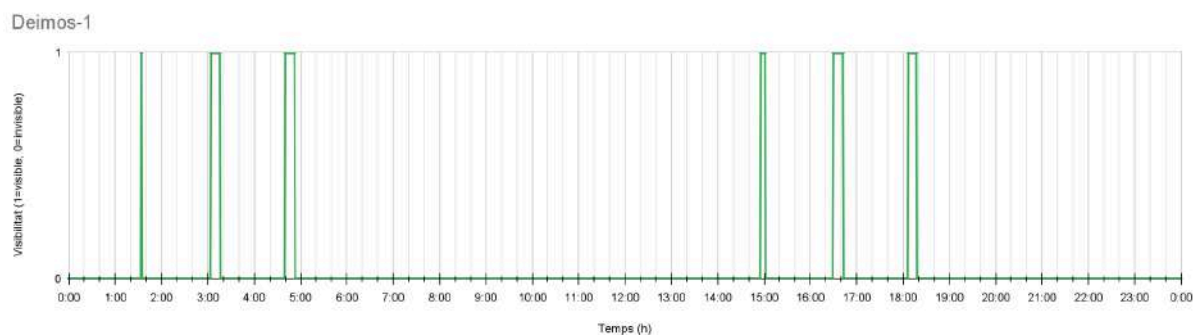


Fig. 47 Gràfica visibilitat (1=visible, 0=invisible)-temps (h) de Deimos-1 (35681) l'1 d'agost de 2025
Font: pròpia

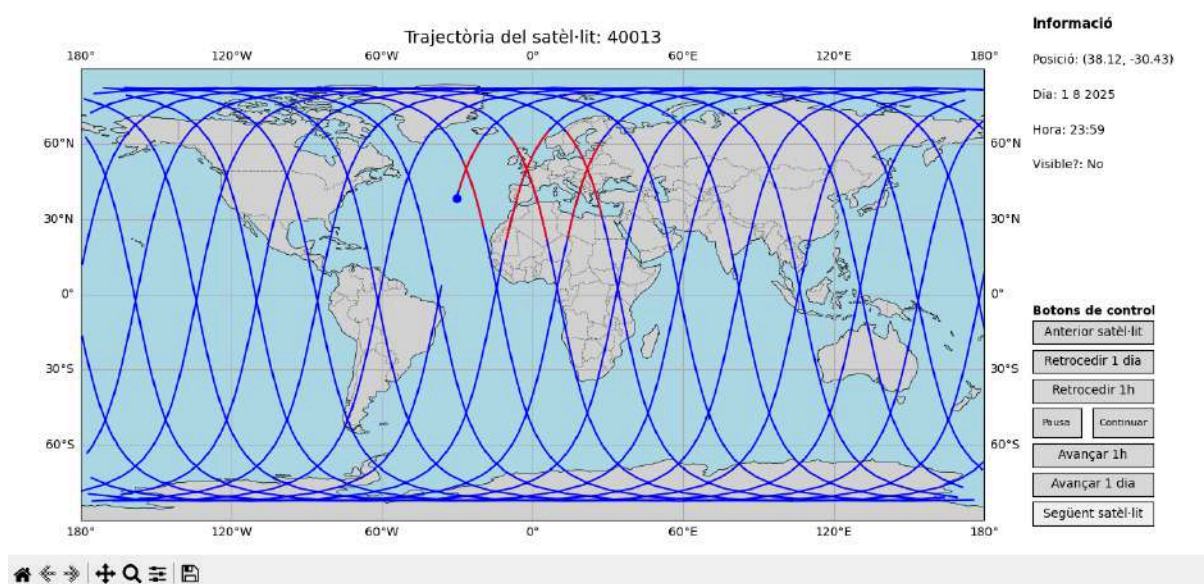


Fig. 48 Predicció de la trajectòria de Deimos-2 (40013) l'1 d'agost de 2025
Font: pròpia

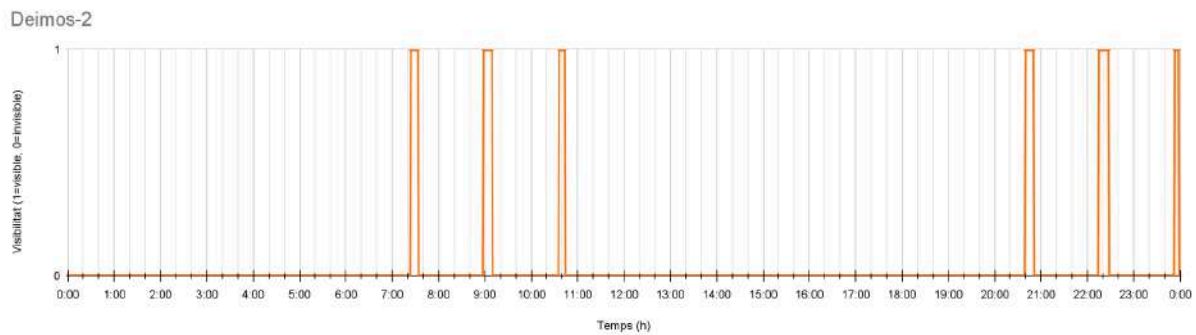


Fig. 49 Gràfica visibilitat (1=visible, 0=invisible)-temps (h) de Deimos-2 (40013) l'1 d'agost de 2025

Font: pròpia

7.5 Starlink-1010 (44716)

Starlink començant al mar de Java, dins de l'oceà Pacífic sud-oest d'Indonèsia (-4,47, 116,77) i acabant a l'oceà Atlàntic, a l'est de la costa de Brasil (-32,28, -38,27), presenta una gran diferència respecte les observades anteriorment. Les corbes també són sinusoidals, però més arrodonides i no passen pels pols, de fet la latitud només se situa entre -53,03 i 53,03 amb aquests valors inclosos, indicant que l'òrbita varia respecte a les altres, no és polar sinó inclinada. Segons els resultats obtinguts, és visible durant 7 períodes, 1 de més a més comparat amb els d'abans a les següents hores: 9:22-9:31 (9 min), 10:57-11:07 (10 min), 12:34-12:43 (9 min), 14:11-14:20 (9 min), 15:47-15:57 (10 min), 17:23-17:33 (10 min), 19:03-19:04 (1 min). Sumades donen 58 min.

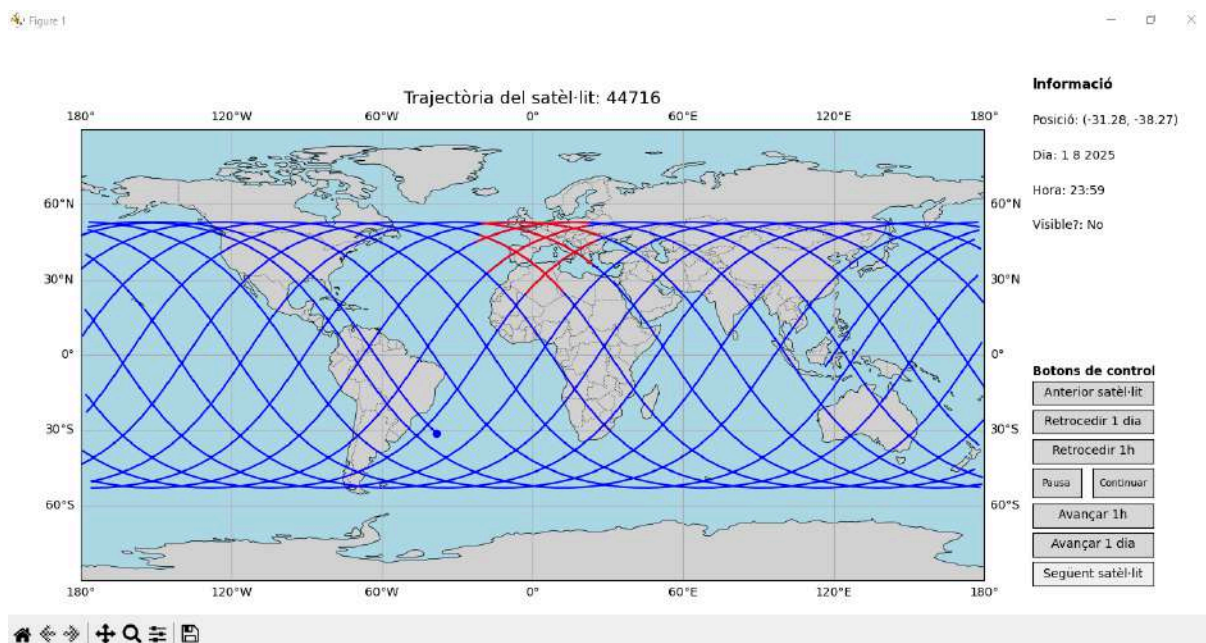


Fig. 50 Predicció de la trajectòria de Starlink-1010 (44716) l'1 d'agost de 2025

Font: pròpia

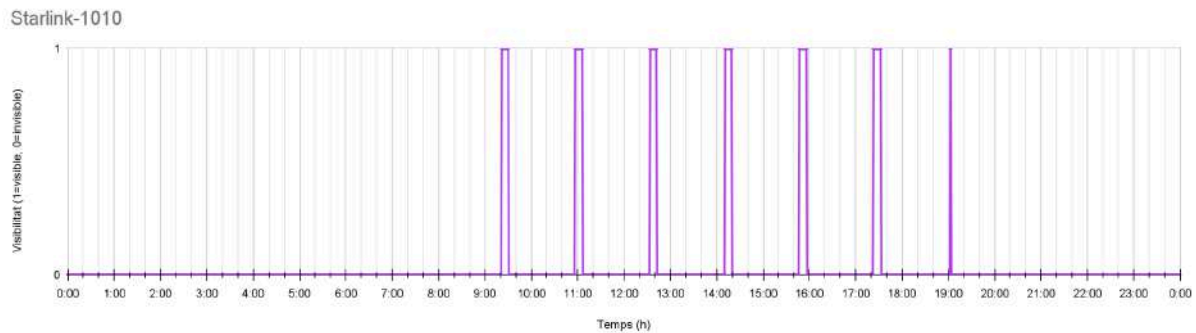


Fig. 51 Gràfica visibilitat (1=visible, 0=invisible)-temps (h) de Starlink-1010 (44716) l'1 d'agost de 2025
Font: pròpia

7.6 Oneweb-0012 (44057)

En darrer lloc, hi ha Oneweb-0012 (44057). Aquesta també és molt diferent de les altres. Comença al sud de l'oceà Atlàntic (-9,41, -17,58), ja visible. Les curvatures en comptes de ser arrodonides són més aviat rectangulars. Es mouen únicament de manera horitzontal cap a la dreta i baixen rectes seguint les línies de latitud. Això és degut al fet que són òrbites polars, ja que passen pels pols. Acaba al nord de l'oceà Atlàntic punt (44,33, -16,12). És visible durant 112 min, 1 h i 5 min, repartits en 8 trams, 00:08-00:26 (16 min), 7:55-8:07 (12 min), 9:42-10:02 (20 min), 11:32-11:50 (18 min), 13:26-13:31 (5 min), 20:16-20:28 (12 min), 22:00-22:19 (19 min), 23:50-00:00 (10 min).

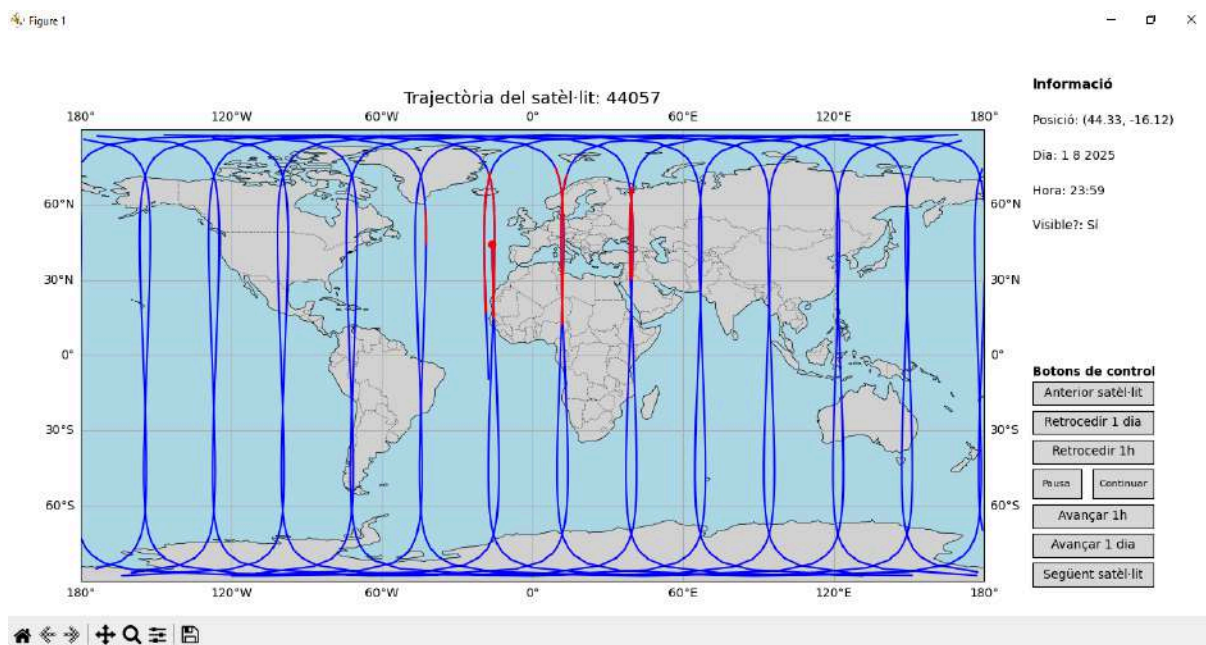


Fig. 52 Predicció de la trajectòria de Oneweb-0012 (44057) l'1 d'agost de 2025
Font: pròpia

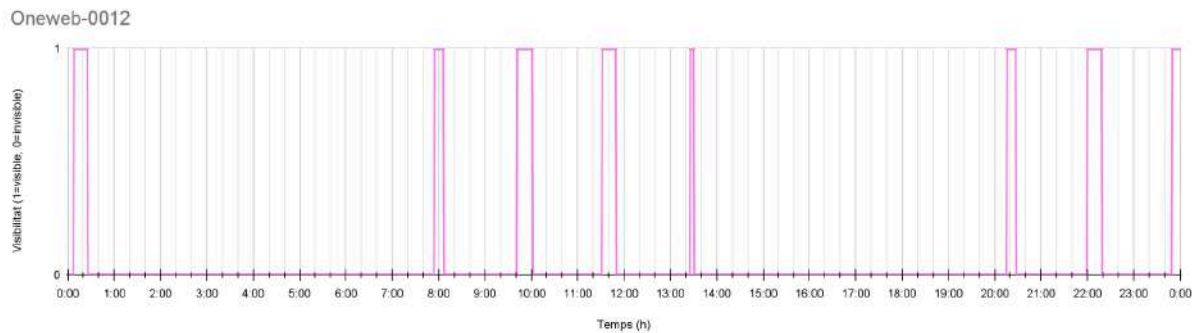


Fig. 53 Gràfica visibilitat (1=visible, 0=invisible)-temps (h) de Oneweb-0012 (44057) l'1 d'agost de 2025
Font: pròpia

7.7 Quantitat de satèl·lits visibles segons el temps a l'1 d'agost de 2025

Si es combinen les taules dels gràfics anteriors se'n deriva un que indica la quantitat de satèl·lits apreciables segons el temps. En el qual es detecta que dos satèl·lits són visibles alhora durant 11 trams. No obstant això, períodes on en són observables tres, que correspon al màxim nombre de satèl·lits que són visibles a la vegada per l'1 d'agost del 2025, es redueixen a tres moments concrets: 9:23-9:28 durant 5 min (LandSat-8, Terra, Starlink-1010), 11:00-11:04 durant 4 min (LandSat-8, Terra, Starlink-1010), 22:07-22:10 durant 3 min (LandSat-8, Terra, Oneweb-0012).

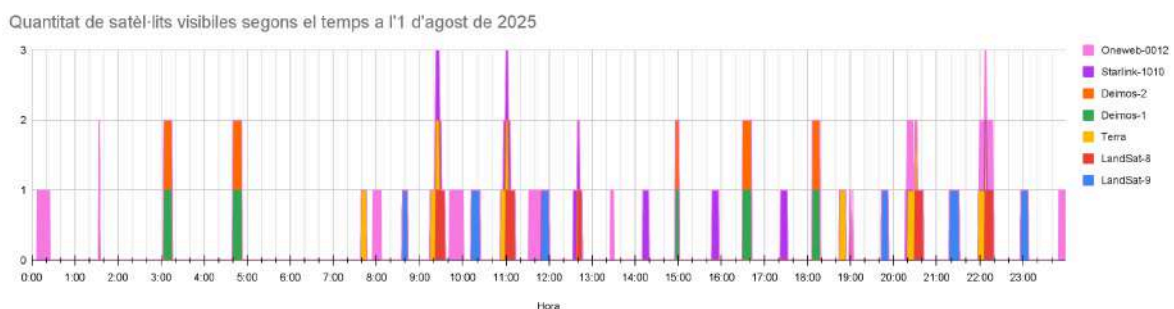


Fig. 54 Gràfica quantitat de satèl·lits visibles-segons el temps a l'1 d'agost de 2025
Font: pròpia

8. EFECTIVITAT DELS RESULTATS OBTINGUTS AL LLARG DEL TEMPS

Per la part pràctica s'han simulat set satèl·lits durant 31 dies. Per comprovar si s'ha pogut predir correctament les posicions, s'agafaran les coordenades del fitxer de sortida de SGP4 obtingudes del TLE del 25 de juliol i es compararan amb els fitxers publicats del moment concret. Ja que 31 dies és un període molt extens s'ha fet una comparació cada cinc dies. D'aquí s'extraurà l'error absolut. Fent l'absolut de la diferència del valor real i el previst, i el relatiu que expressa la diferència en relació amb el valor real. Només s'ha realitzat pel primer satèl·lit perquè en ser gestionats pel mateix simulador l'error serà l'idèntic i també per la llargada limitada del treball. Des de la pàgina web de Celestrak es van sol·licitar tots els fitxers TLE dintre el període d'entre el 25 de juliol fins al 24 d'agost.

8.1 30 de juliol

Aquestes dues línies no ens poden proporcionar la posició de manera directa pel que s'ha agafat el simulador orbital, s'ha reemplaçat els fitxers TLE pels nous i s'ha modificat la data d'iniciació per 5 dies més tard, en aquest cas el 30 de juliol i la durada per 1 dia, pel fet que no són necessaris més dies.

```
# Data per a la qual volem fer el càlcul
data_any = 2025
data_mes = 7
data_dia = 30
data_hora = 00
data_min = 00
data_segons = 00

# Interval de temps per al qual volem calcular les òrbites
num_days = 1
num_hours = 24
num_min = 60
```

Fig. 55 Modificació de paràmetres. La data d'iniciació per 30/07/2025 i la duració per 1 dia.
Font: pròpia

```
main.py  gps-all-sats.tle  dadesGPS.txt
tlfFiles > gps-all-sats.tle
1 LEO LANDSAT9 (ID 49260+5dia)
2 1 49260U 21088A 25211.48014438 .00000365 00000-0 91036-4 0 9990
3 2 49260 98.2253 281.4227 0001314 84.6500 275.4848 14.57111390204141
4 |
```

Fig. 56 Introducció de TLE de LandSat-9 del 30 de juliol a les 11:31
Font: pròpia

Es consulta el quart component que ens indica que la data de les dades, en aquest cas, correspon el dia 211è de l'any, que és el 30 de juliol amb la fracció del dia 0,48014438 del dia. Per saber el moment exacte es converteixen a hores i minuts, es multiplica la fracció pel total d'hores del dia i es passa la part decimal a minuts.

$$0,48014438 \text{ dia} \times \frac{24 \text{ h}}{1 \text{ dia}} = 11,523 \text{ h}$$

$$0,523 \text{ h} \times \frac{60 \text{ min}}{1 \text{ h}} = 31,41 \text{ min}$$

S'obté que són dades de les 11:31 i es busca en el fitxer de sortida la posició durant aquest mateix instant:

```
main.py  gps-all-sats.tle  dadesGPS.txt
txtFiles > dadesGPS.txt
689 49260+5dia 0 30 7 2025 11:28 -71.8 (deg) 35.9 (deg) -6615.9690259 2039.4214268 -1506.5340993
690 49260+5dia 0 30 7 2025 11:29 -69.9 (deg) 34.5 (deg) -6663.8585759 2155.1155230 -1068.6512181
691 49260+5dia 0 30 7 2025 11:30 -68.0 (deg) 33.2 (deg) -6683.9871445 2262.4447944 -626.4412313
692 49260+5dia 0 30 7 2025 11:31 -66.1 (deg) 32.2 (deg) -6676.3418649 2360.7321064 -181.6939606
693 49260+5dia 0 30 7 2025 11:32 -64.2 (deg) 31.2 (deg) -6641.0283230 2449.3372684 263.7896839
```

Fig. 57 Línia del fitxer de sortida amb les coordenades de posició per LandSat-9 el 30 de juliol a les 11:31
Font: pròpia

Tanmateix, com que les dades es troben en coordenades ECEF, les hem de transformar a longitud i latitud, ja que són amb les quals s'ha estat treballant durant tot el projecte. S'apliquen altra vegada les fórmules de l'apartat 6.3.3.

$$latitude = \arcsin\left(\frac{-181.694}{\sqrt{(-6676.341)^2 + (2360.732)^2 + (-181.694)^2}}\right) = -1.47^\circ$$

$$longitude = \text{atan2}(2360.732, -6676.341) = 160.53^\circ$$

Si s'observa la prèvia predicció, es veu que hauria d'estar a (-1,48, 160,53) mentre que a la conversió ens diu que a la vida real es va trobar a (-1,47, 160,53). Amb l'error absolut, fent l'absolut de la diferència dels dos valors, s'ha vist que només presenta una petita diferència de 0,01° a la latitud, però per fer-ho més entenedor s'ha convertit en una de les unitats que es fa servir de manera habitual, el quilòmetre. Es parteix de l'equivalència 1°=111 km i es faria un factor de conversió amb el qual s'aconsegueix que són 1,11 km.

$$|-1,47 - (-1,48)| = 0,01^\circ$$

$$0,01^\circ \times \frac{111\text{km}}{1^\circ} = 1,11\text{km}$$

En termes relatius, l'error és mínim. Es tracta únicament del 0,68% del valor original de la latitud.

$$\frac{0,01}{1,47} \times 1,47 = 0,68\%$$

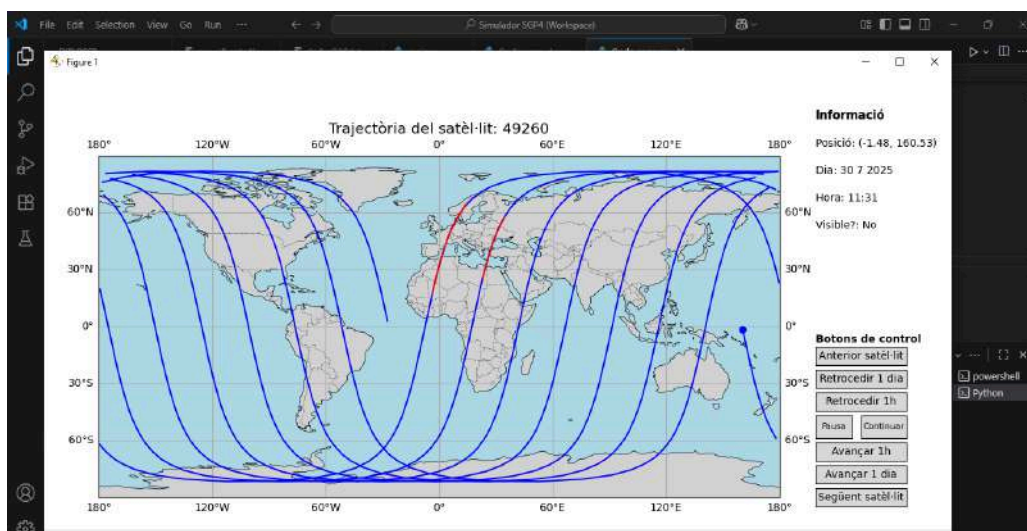


Fig. 58 Visualització des de l'animació creada la posició prevista de LandSat-9 del 30 de juliol a les 11:31
Font: pròpia



Fig. 59 Visualització des de Google Maps la posició real de LandSat-9 del 30 de juliol a les 11:31
Font: <https://www.google.com/maps/>

8.2 4 d'agost

Per cada cas el procés anterior es repetirà, però adaptant-se a les noves dades. Es canvia la data d'iniciació per 5 dies més tard respecte a l'anterior, que són 10 posteriors al 25 de juliol, el 4 d'agost i de seguida s'incorpora el nou fitxer TLE.

```
10 data_any = 2025
11 data_mes = 8
12 data_dia = 4
13 data_hora = 00
14 data_min = 00
15 data_seg = 00
```

Fig. 60 Modificació de paràmetres. La data d'iniciació és canviada per 04/08/2025
Font: pròpia

```
main.py  gps-all-sats.tle  dadesGPS.txt
tleFiles > gps-all-sats.tle
1 LEO LANDSAT9 [ID 49260+10dia]
2 1 49260U 21088A 25216.21838258 .00000326 00000+0 82475-4 0 9993
3 2 49260 98.2248 286.1016 0001196 88.9739 271.1597 14.57094346204844
```

Fig. 61 Introducció de TLE de LandSat-9 del 4 d'agost a les 5:14
Font: pròpia

Aquest indica que és del dia 216è de l'any, el 4 d'agost i a la fracció del dia 0,21838258. Es tornaran a aplicar els factors de conversió:

$$0,21838258 \text{ dia} \times \frac{24 \text{ h}}{1 \text{ dia}} = 5,2412 \text{ h}$$

$$0,2412 \text{ h} \times \frac{60 \text{ min}}{1 \text{ h}} = 14,47 \text{ min}$$

Amb els quals s'obté 5:14 com a hora.

```
49260+10dia 0 4 8 2025 05:13 -50.8 (deg) 277.7 (deg) -1755.3166524 -6832.2433531 -654.3785875
49260+10dia 0 4 8 2025 05:14 -49.8 (deg) 281.0 (deg) -1854.3551148 -6833.6820825 -209.7436880
49260+10dia 0 4 8 2025 05:15 -48.8 (deg) 284.2 (deg) -1945.8250587 -6806.8058588 235.7408712
49260+10dia 0 4 8 2025 05:16 -47.7 (deg) 287.4 (deg) -2020.1008847 -6751.7856268 680.2701432
```

Fig. 62 Línia del fitxer de sortida amb les coordenades de posició per LandSat-9 el 4 d'agost a les 5:14
Font: pròpia

Tot seguit les coordenades ECEF són transformades i s'aconsegueix que està a (-1,70, -105,18) en la realitat.

$$latitude = \arcsin\left(\frac{-209.744}{\sqrt{(-1854.36)^2 + (-6833.68)^2 + (-209.744)^2}}\right) = -1.70^\circ$$

$$longitude = \text{atan2}(-6833.68, -1854.36) = -105.18^\circ$$

En comparar-les amb la posició prevista (-1,47, -105,22) es detecta una diferència de 0,23° en la latitud i de 0,04° en la longitud que és l'error absolut i que equivalen a 25,53 km i 4,44 km respectivament.

$$|-1.70 - (-1.47)| = 0.23^\circ$$

$$|-105.18 - (-105.22)| = 0.04^\circ$$

$$0.23^\circ \times \frac{111\text{km}}{1^\circ} = 25.53\text{km}$$

$$0.04^\circ \times \frac{111\text{km}}{1^\circ} = 4.44\text{km}$$

Si es calcula el relatiu en la latitud arriba fins al 13,53% del valor original i per la longitud fins a un 0,04%.

$$\frac{0.23}{1.70} \times 100 = 13.53\%$$

$$\frac{0.04}{105.18} \times 100 = 0.04\%$$

S'ha observat que contraposat el primer escenari els errors han incrementat, un 12,85%, 24,42 km per la latitud que és bastant significatiu i un 0,04%, 4,44 km per la longitud, pràcticament invariable. Aplicant el teorema de Pitàgores s'ha calculat la distància total del punt previst i el real que ha ocasionat 25,91 km. La conclusió implica que en només 5 dies, ha augmentat 24,8 km més que abans. Tot i que si es comparen visualment les dues posicions la diferència no és gens evident. L'error encara és acceptable en termes de visibilitat. El satèl·lit encara es mantindria en la zona esperada, pel que no seria un problema.

$$\sqrt{25.53^2 + 4.44^2} = 25.91\text{ km}$$

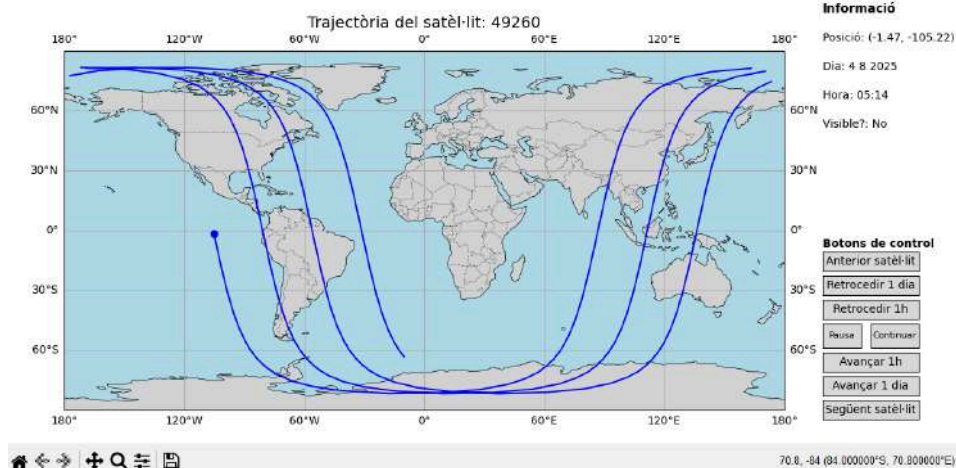


Fig. 63 Visualització des de l'animació creada la posició prevista de LandSat-9 del 4 d'agost a les 5:14
Font: pròpia



Fig. 64 Visualització des de Google Maps la posició real de LandSat-9 del 4 d'agost a les 5:14
Font: <https://www.google.com/maps/>

8.3 9 d'agost

Per aquesta tercera circumstància, s'agafa el fitxer TLE de 15 dies més tard, el 221è de l'any que és el 9 d'agost. També es canvia la data d'iniciació del simulador per aquest mateix dia.

```
10 data_any = 2025
11 data_mes = 8
12 data_dia = 9
13 data_hora = 00
14 data_min = 00
15 data_segons = 00
16
```

Fig. 65 Modificació de paràmetres. La data d'iniciació és canviada per 09/08/2025
Font: pròpia

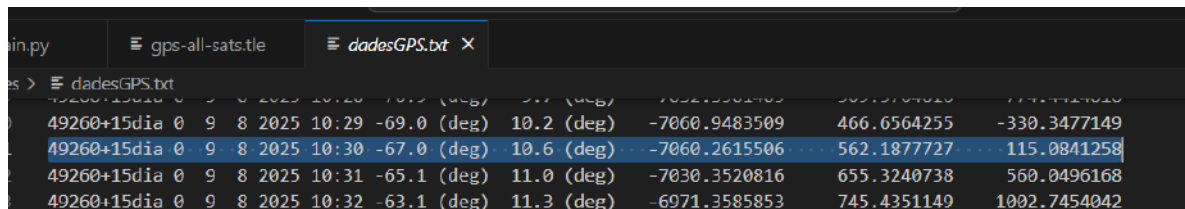
```
ain.py  gps-all-sats.tle  dadesGPS.txt
es > gps-all-sats.tle
LEO LANDSAT9 (ID 49260+15dia)
1 49260U 21088A 25221.43732070 .00000351 00000+0 88063-4 0 9995
2 49260 98.2246 291.2550 0001150 90.4993 269.6338 14.57098502205598
```

Fig. 66 Introducció de TLE de LandSat-9 del 9 d'agost a les 5:14
Font: pròpia

A posteriori, es busca en el document de sortida la posició durant les 10:30 que s'ha determinat després de dur a terme les operacions de conversió de fracció del dia a hores i minuts.

$$0,43732070 \text{ dia} \times \frac{24 \text{ h}}{1 \text{ dia}} = 10,496 \text{ h}$$

$$10,496 \text{ h} \times \frac{60 \text{ min}}{1 \text{ h}} = 29,74 \text{ min}$$



Line	Time	Lat (deg)	Long (deg)	Alt (m)	Speed (m/s)	Course (deg)	Fix Quality	PDOP	HDOP	VDOP	SDOP
49260+15dia 0 9 8 2025 10:29	10:29	-69.0	10.2	-7060.9483509	466.6564255	-330.3477149	1	1.2	1.2	1.2	1.2
49260+15dia 0 9 8 2025 10:30	10:30	-67.0	10.6	-7060.2615506	562.1877727	115.0841258	1	1.6	1.6	1.6	1.6
49260+15dia 0 9 8 2025 10:31	10:31	-65.1	11.0	-7030.3520816	655.3240738	560.0496168	1	1.0	1.0	1.0	1.0
49260+15dia 0 9 8 2025 10:32	10:32	-63.1	11.3	-6971.3585853	745.4351149	1002.7454042	1	1.3	1.3	1.3	1.3

Fig. 67 Línia del fitxer de sortida amb les coordenades de posició per LandSat-9 el 8 d'agost a les 10:30
Font: pròpia

Les coordenades ECEF es transformen en latitud i longitud que resulten 0,93° i 175,45°.

$$\text{latitude} = \arcsin\left(\frac{115.08}{\sqrt{(-7060.26)^2 + (562.19)^2 + (115.08)^2}}\right) = 0.93^\circ$$

$$\text{longitude} = \text{atan2}(562.18, -7060.26) = 175.45^\circ$$

S'havia previst que fossin a (1,54, 175,36), però la realitat no ha concordat amb ella. Un cop fet l'error absolut s'ha vist que la diferència és de -0,61° i 0,09°. Convertint-les a distàncies s'obté que són de 67,71 km i 9,99 km.

$$|0,93 - (1,54)| = 0,61^\circ$$

$$|175,45 - (175,36)| = 0,09^\circ$$

$$0,61^\circ \times \frac{111 \text{ km}}{1^\circ} = 67,71 \text{ km}$$

$$0,09^\circ \times \frac{111 \text{ km}}{1^\circ} = 9,99 \text{ km}$$

L'error relatiu en la latitud ha superat la meitat del valor original, amb un 65,59%, ha incrementat 52,06%, 42,18 km que el segon cas, mentre que la longitud continua essent molt petit, només un 0,05% augmentant un 0,01%, 5,55 km en comparació amb l'anterior situació.

$$\frac{0,61}{0,93} \times 100 = 65,59\%$$

$$\frac{0,09}{175,45} \times 100 = 0,05\%$$

Després d'aplicar Pitàgores es conclou que el punt real està a 68,44 km del previst, 42,53 km més que abans.

$$\sqrt{67,71^2 + 9,99^2} = 68,44 \text{ km}$$

L'error comença a ser considerable.

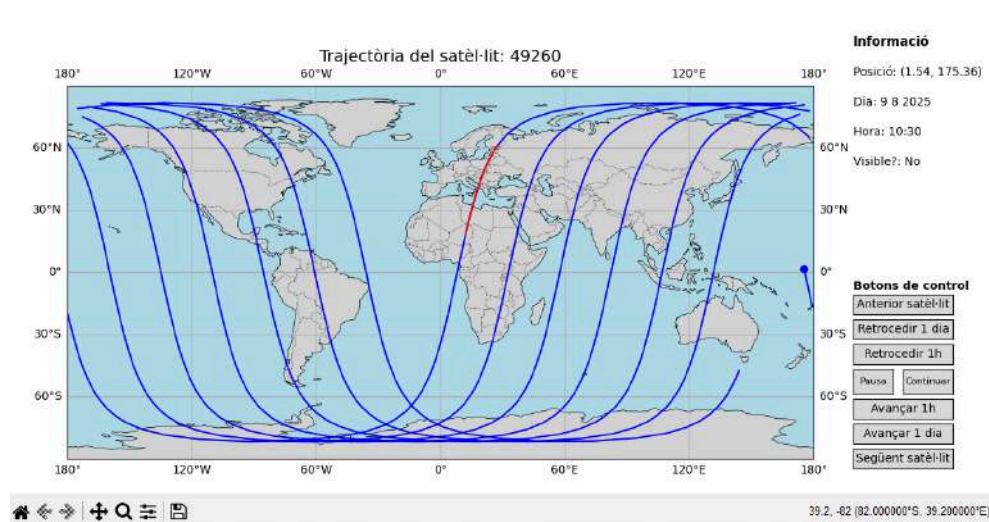


Fig. 68 Visualització des de l'animació creada la posició prevista de LandSat-9 del 8 d'agost a les 10:30
Font: pròpia



Fig. 69 Visualització des de Google Maps la posició real de LandSat-9 del 8 d'agost a les 10:30
Font: <https://www.google.com/maps/>

8.4 14 d'agost

Per fer l'anàlisi s'agafa les dades del dia 226è de l'any, el 14 d'agost i s'adapta el simulador orbital.

```

10 data_any = 2025
11 data_mes = 8
12 data_dia = 14
13 data_hora = 00
14 data_min = 00
15 data_segons = 00
16

```

Fig. 70 Modificació de paràmetres.
La data d'iniciació és canviada
per 14/08/2025
Font: pròpia

```

gps-all-sats.tle x  gps-all-sats.txt
> gps-all-sats.tle
LEO LANDSAT9 (ID 49260+20dia)
1 49260U 21088A 25226.45023278 .00000359 00000+0 89849-4 0 9996
2 49260 98.2240 296.2045 0001147 92.8913 267.2417 14.57102949206336

```

Fig. 71 Introducció de TLE de LandSat-9 del 14 d'agost a les 10:48
Font: pròpia

El fitxer TLE proporcionat és del moment 0,45023278 del dia, si es passa en hores, equival a 10,806. Per saber els minuts s'han utilitzat els decimals i se'ls ha multiplicat per 60 i s'ha obtingut 48,335 minuts. Per tant, a partir d'aquí es busca dins del fitxer de sortida la línia que correspon a les 10:48 i s'extrauran les coordenades.

$$0,45023278 \text{ dia} \times \frac{24 \text{ h}}{1 \text{ dia}} = 10,806 \text{ h}$$

$$10,806 \text{ h} \times \frac{60 \text{ min}}{1 \text{ h}} = 48,335 \text{ min}$$

49260+20dia 0 14	8	2025	10:46	-71.6 (deg)	17.4 (deg)	-6953.5826438	876.4459709	-1036.6006041
49260+20dia 0 14	8	2025	10:47	-69.7 (deg)	17.3 (deg)	-6991.3595285	977.8509329	-594.1524938
49260+20dia 0 14	8	2025	10:48	-67.7 (deg)	17.2 (deg)	-7000.1564355	1075.5281334	-149.2982763
49260+20dia 0 14	8	2025	10:49	-65.8 (deg)	17.2 (deg)	-6979.9662752	1168.8302004	296.1607386
49260+20dia 0 14	8	2025	10:50	-63.8 (deg)	17.1 (deg)	-6930.9052497	1257.1277607	740.4196238

Fig. 72 Línia del fitxer de sortida amb les coordenades de posició per LandSat-9 el 14 d'agost a les 10:48
Font: pròpia

La latitud i la longitud es determinen amb les operacions:

$$\text{latitude} = \arcsin\left(\frac{-149.3}{\sqrt{(-7000.16)^2 + (1075.53)^2 + (-149.3)^2}}\right) = -1.21^\circ$$

$$\text{longitude} = \text{atan2}(1075.52, -7000.15) = 171.27^\circ$$

El resultat és comparat amb les xifres reals amb el qual s'obté que presenta una diferència de 0,97° i 0,15° respectivament. Així que en la latitud es desvia uns 107,67 km de la ubicació de la realitat. Calculant el percentatge d'error es té 0,97 sobre -1,21, dada que representa un 80,17%. Pel que fa a la longitud presenta un desfasament de 16,65 km, que suposa una inexactitud de 0,05%.

$$|-1,21 - (-0,24)| = 0,97^\circ$$

$$|171,27 - (171,12)| = 0,15^\circ$$

$$0,97^\circ \times \frac{111 \text{ km}}{1^\circ} = 107,67 \text{ km}$$

$$0,15^\circ \times \frac{111 \text{ km}}{1^\circ} = 16,65 \text{ km}$$

$$\frac{0,97}{-1,21} \times 100 = 80,17\%$$

$$\frac{0,15}{171,27} \times 100 = 0,09\%$$

Finalment, combinant les dues desviacions com si fossin els catets d'un triangle rectangle la distància la discrepància total entre les dades simulades i les observades és d'uns 108,95 km.

$$\sqrt{107,67^2 + 16,65^2} = 108,95 \text{ km}$$

Si es recapitula, es veu que per un costat, la latitud la diferència passa de 0,61° a 0,97°, és a dir, que augmenta 0,36°, uns 39,96 km. El percentatge donat també ha pujat, passa de 65,59% a 80,17%, per tant, presenta un increment de 14,57%. Per l'altre costat, la longitud de 0,09° va a 0,15°, que és un increment de 0,06°, 6,66 km. L'error relatiu també ha seguit els passos dels altres, ocasionant una diferència de 0,04%. Els quilòmetres totals combinats augmenten en 40,51 km.

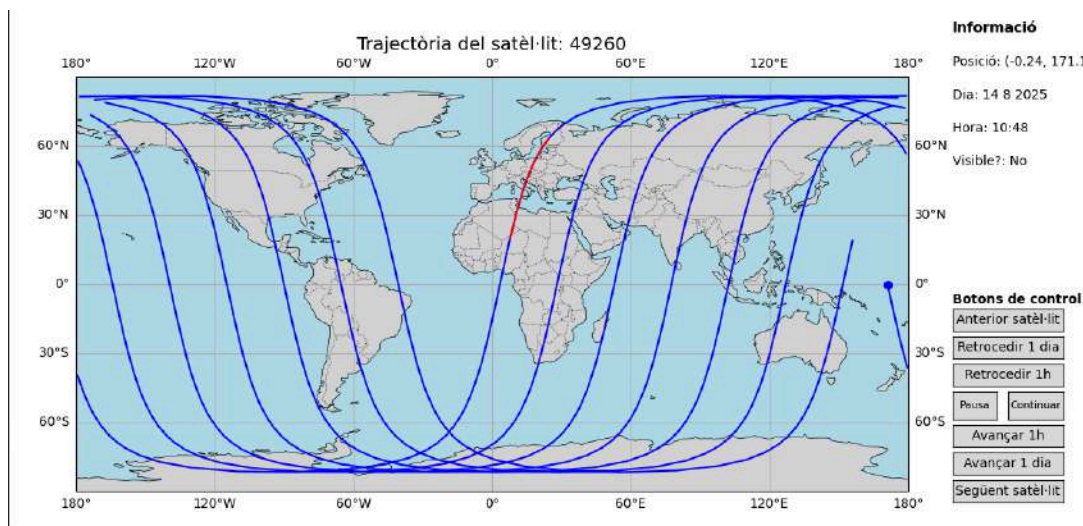


Fig. 73 Visualització des de l'animació creada la posició prevista de LandSat-9 del 14 d'agost a les 10:48
Font: pròpia



Fig. 74 Visualització des de Google Maps la posició real de LandSat-9 del 14 d'agost a les 10:48

Font: <https://www.google.com/maps/>

8.5 19 d'agost

Novament, es renova el fitxer TLE amb les dades actualitzades. Addicionalment, es modifica la data del codi pel 19 d'agost, que és el dia 231è de l'any.

```
data_any    = 2025
data_mes    = 8
data_dia     = 19
data_hora    = 00
data_min     = 00
data_segons  = 00
```

Fig. 75 Modificació de paràmetres. La data d'iniciació és canviada per 19/08/2025
Font: pròpia

```
main.py  dadesGPS.txt  gps-all-sats.tle X
es > gps-all-sats.tle
LEO LANDSAT9 [ID 49260+25dia]
1 49260U 21088A 25231.18845242 .00000304 00000+0 77494-4 0 9998
2 49260 98.2234 300.8825 0001120 95.7894 264.3432 14.57106382207025
```

Fig. 76 Introducció de TLE de LandSat-9 del 19 d'agost a les 4:31
Font: pròpia

En fer els càlculs de conversió, es comprova que la informació fou publicada a les 4.31 del matí. S'ha obtingut a partir del valor 0,18845242 dies, que s'ha extret del quart component del TLE. En passar-ho a hores multiplicant per 24, s'aconsegueix 4,523 hores. Els decimals són multiplicats per 60 i això dona 31,37 minuts.

$$0,18845242 \text{ dia} \times \frac{24 \text{ h}}{1 \text{ dia}} = 4,523 \text{ h}$$

$$0,523 \text{ h} \times \frac{60 \text{ min}}{1 \text{ h}} = 31,37 \text{ min}$$

49260+25dia 0 19 8 2025 04:30 -46.4 (deg) 270.6 (deg)	-453.6648686	-7043.4866237	-610.2614309
49260+25dia 0 19 8 2025 04:31 -45.4 (deg) 273.9 (deg)	-550.4854977	-7060.5580785	-165.4645140
49260+25dia 0 19 8 2025 04:32 -44.4 (deg) 277.1 (deg)	-645.1138608	-7048.3980785	280.0026804
49260+25dia 0 19 8 2025 04:33 -43.4 (deg) 280.2 (deg)	-736.9133456	-7007.0713766	724.3352416

Fig. 77 Línia del fitxer de sortida amb les coordenades de posició per LandSat-9 el 19 d'agost a les 4:31
Font: pròpia

$$\text{latitude} = \arcsin\left(\frac{-149.3}{\sqrt{(-7000.16)^2 + (1075.53)^2 + (-149.3)^2}}\right) = -1.34^\circ$$

$$\text{longitude} = \text{atan2}(1075.52, -7000.15) = -94.46^\circ$$

Per saber quant es desvia s'aplica la fórmula de l'error absolut. Entre la latitud prevista i la real hi ha una diferència d'1,32°. En la longitud és de 0,19°.

$$|-1,34 - (-0,02)| = 1,32^\circ$$

$$|-94,46 - (-94,65)| = 0,19^\circ$$

Per transformar aquestes diferències angulars en distàncies sobre la superfície terrestre es pren com a referència que 1 grau equival a aproximadament 111 km. Aleshores, 1,32° equival a 146,52 km i 0,19° a 21,09 km. Quan es parla en termes percentuals, aquest primer representa el 98,51% del valor de la realitat, vol dir que el resultat serà de baixa fiabilitat, presenta un augment de 18,34% respecte a l'anterior circumstància. El segon continua sent un desajust molt petit de 0,2%, en comparació en l'última situació ha incrementat un 0,11%.

$$1,32^\circ \times \frac{111\text{km}}{1^\circ} = 146,52\text{km}$$

$$0,19^\circ \times \frac{111\text{km}}{1^\circ} = 21,09\text{km}$$

$$\left| \frac{1,32}{-1,34} \right| \times 100 = 98,51\%$$

$$\left| \frac{0,19}{-94,46} \right| \times 100 = 0,2\%$$

Entre la posició real i la prevista hi haurà una distància de 148,03 km, que són 39,08 km si és comparat amb la circumstància d'abans.

$$\sqrt{146,52^2 + 21,09^2} = 148,03 \text{ km}$$

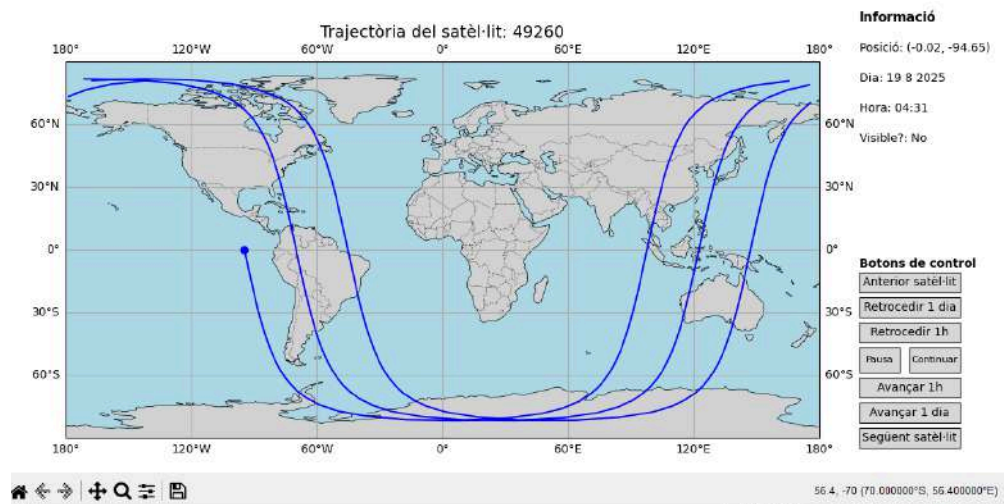


Fig. 78 Visualització des de l'animació creada la posició prevista de LandSat-9 del 19 d'agost a les 4:31

Font: pròpia



Fig. 79 Visualització des de Google Maps la posició real de LandSat-9 del 19 d'agost a les 4:31

Font: <https://www.google.com/maps/>

8.6 24 d'agost

Trenta dies més tard respecte al 25 de juliol és el 24 d'agost. El fitxer TLE que s'ha buscat per aquest dia es veu que coincideix amb el dia introduït en el simulador, marcant que és del 236è dia de l'any.

```
data_any = 2025
data_mes = 8
data_dia = 24
data_hora = 00
data_min = 00
data_segons = 00
```

Fig. 80 Modificació de paràmetres. La data d'iniciació és canviada per 24/08/2025.
Font: pròpia

```
main.py dadesGPS.txt gps-all-sats.tle
les > gps-all-sats.tle
LEO LANDSAT9 (ID 49260+30dia)
1 49260U 21088A 25236.61335735 .00000322 00000+0 81506-4 0 9991
2 49260 98.2228 306.2379 0001120 98.7181 261.4144 14.57110291207813
```

Fig. 81 Introducció de TLE de LandSat-9 del 24 d'agost a les 14:43
Font: pròpia

Per identificar el moment exacte en què s'han obtingut s'han efectuat unes operacions, primer per saber l'hora i després per conèixer els minuts. Com a resultat s'ha obtingut que foren aconseguides a les 14:43.

$$0,61335735 \text{ dia} \times \frac{24 \text{ h}}{1 \text{ dia}} = 14,721 \text{ h}$$

$$0,721 \text{ h} \times \frac{60 \text{ min}}{1 \text{ h}} = 43 \text{ min}$$

```
s > dadesGPS.txt
49260+30dia 0 24 8 2025 14:42 -51.9 (deg) 79.3 (deg) -2788.2265738 6489.3949168 -549.4749947
49260+30dia 0 24 8 2025 14:43 -50.2 (deg) 77.1 (deg) -2708.1720262 6544.9352971 -104.4811991
49260+30dia 0 24 8 2025 14:44 -48.6 (deg) 75.0 (deg) -2616.8587917 6573.4051231 340.9361267
```

Fig. 82 Línia del fitxer de sortida amb les coordenades de posició per LandSat-9 el 24 d'agost a les 14:43
Font: pròpia

Les coordenades ECEF són transformades en latitud i longitud i comparant amb els valors que s'han aconseguit amb les dades de referència, la diferència absoluta del primer cas és d'1,74°, 193,14 km i del segon és de 0,25°, 27,75 km.

$$\text{latitude} = \arcsin\left(\frac{-104.48}{\sqrt{(-2708.17)^2 + (6544.94)^2 + (-104.48)^2}}\right) = -0,85^\circ$$

$$\text{longitude} = \text{atan2}(6544.94, -2708.17) = 112,48^\circ$$

$$|-0,85 - (0,89)| = 1,74^\circ$$

$$|112,48 - (112,23)| = 0,25^\circ$$

$$1,74^\circ \times \frac{111 \text{ km}}{1^\circ} = 193,14 \text{ km}$$

$$0,25^\circ \times \frac{111 \text{ km}}{1^\circ} = 27,75 \text{ km}$$

Aleshores calculat el percentatge d'error respecte al valor total real. Amb els quals han donat com a resultat 204,71% per la latitud, que es desvia més del doble de la xifra de la

coordenada legítima, per aquest motiu no és fiable. Ha augmentat un 106,2% si es compara amb l'última situació. Per la longitud, continua sent petita, ha augmentat únicament 0,02%. Tenint en compte els dos valors s'obté que presenta que la previsió presenta un desviament de 195,12 km.

$$\left| \frac{1,74}{-0,85} \right| \times 100 = 204,71\%$$

$$\left| \frac{0,25}{112,48} \right| \times 100 = 0,22\%$$

$$\sqrt{193,14^2 + 27,75^2} = 195,12 \text{ km}$$

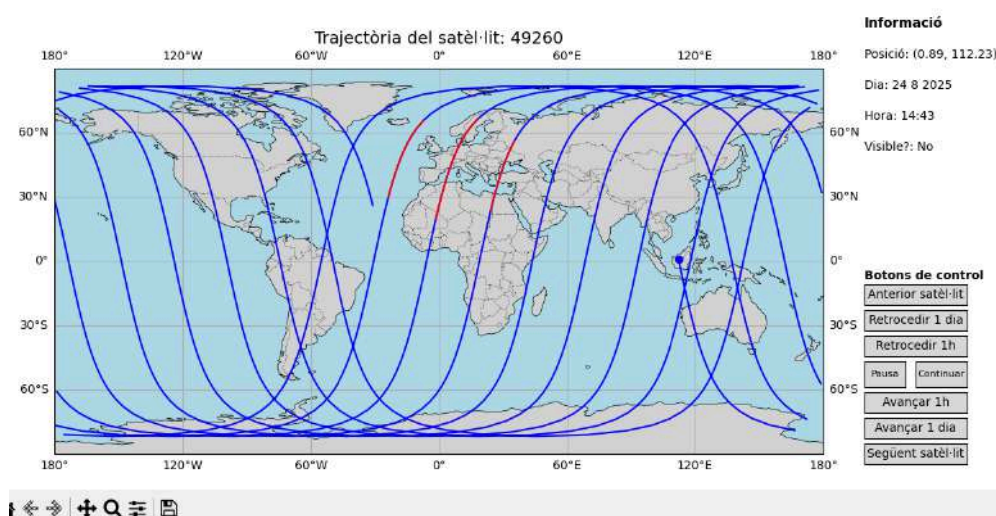


Fig. 83 Visualització des de l'animació creada la posició prevista de LandSat-9 del 24 d'agost a les 14:43
Font: pròpia



Fig. 84 Visualització des de Google Maps la posició real de LandSat-9 del 24 d'agost a les 14:43
Font: <https://www.google.com/maps/>

8.7 Recopilació de resultats i interpretació

Data	30/7/25	4/8/25	9/8/25	14/8/25	19/8/25	24/8/25
Error absolut de latitud (°)	0,01	0,23	0,61	0,97	1,32	1,74
Error absolut de latitud (km)	1,11	25,53	67,71	107,67	146,52	193,14
Error relatiu de latitud (%)	0,68	13,53	65,59	80,17	98,51	204,71
Error absolut de longitud (°)	0	0,04	0,09	0,15	0,19	0,25
Error absolut de longitud (km)	0	4,44	9,99	16,65	21,09	27,75
Error relatiu de longitud (%)	0	0,04	0,05	0,09	0,2	0,22
Distància errònia total (km)	1,11	25,91	68,44	108,95	148,03	195,12

Per analitzar patrons d'augment d'errors s'ha hagut de basar en aproximacions, degut al fet que els fitxers TLE són obtinguts durant hores qualssevol del dia, i en aquest cas cap ha estat de les 00:00. Així doncs, per l'error absolut de la latitud es veu que a partir del segon cas augmenta de manera quasi lineal afegint 0,3-0,4° cada 5 dies, que són uns 40-45 km en superfície. En la longitud, l'increment és molt més lent, s'afegeixen uns 0,04-0,06° cada 5 dies, uns 5-7 km. Això queda reflectit també a la distància total, i s'incrementa a partir de la segona situació uns 40-50 km cada cinc dies. Pel que fa a l'error relatiu en la latitud, s'ha observat un creixement molt ràpid, passant de menys d'1% fins a més de 200% en 30 dies. Ja ens indica que la predicció serà poc fiable a llarg termini.

Durant els primers 5 dies es veu que té un error de només 1,11 km, tenint en compte que el cercle de cobertura dels satèl·lits LEO oscil·la entre 1000 i 2500 km de radi, el satèl·lit apareixerà pràcticament on s'esperava. 10 dies més tard, quan és de 25,91 km, l'error encara és considerat petit, però pot presentar un lleuger desplaçament perceptible. En la majoria dels casos serà visible des del punt de referència. Al cap de 15 dies, l'error ja arriba a 68,44 km. És considerable, pot ser poc fiable per observacions locals i la posició estarà significativament desplaçada, més amunt, a baix o amb una durada diferent. 20 dies més tard, passa a ser de 108,95 km. Aquí l'error ja és molt gran, en comptes de ser a Banyoles, podria ser a Barcelona. Després de 25 dies, l'error és de 148,03 km i després de 30 és de 195,12 km, el satèl·lit estaria situat a Tarragona o algun lloc amb la mateixa distància de Banyoles. No obstant això, no són tan grans com per passar d'un país a un altre, si es compara el mapa amb la ubicació real contra la prevista, es veu que quasi no canvien, és

difícil d'apreciar la diferència. Gràcies a la cobertura dels satèl·lits, en principi els que han sigut considerats com a visibles, ho continuarien sent, ja que no superen una diferència de 1000 km. El que passaria és que potser s'observarien en hores diferents de les previstes, i la ubicació exacta seria diferent. Ara, cal recalcar que l'àrea que es considera observable des de l'institut Pere Alsius és molt àmplia, a vegades també són considerats com a visibles quan estan a quilòmetres més enllà del punt de referència, perquè es considera com a apreciable si està per sobre de l'horitzó. Aleshores, teòricament serà visible i en la realitat hi haurà la possibilitat que pugui ser vist des dels punts marcats en vermell; no obstant això, només ho serà realment si no hi ha obstacles en la línia de visió directa entre l'observador i el satèl·lit.

9. CONCLUSIONS

En primer lloc, s'ha pogut validar la primera hipòtesi: “És possible visualitzar les trajectòries dels satèl·lits en mapes 2D amb els quals és possible veure on estaran a mesura que recorri el temps durant 31 dies (1 mes)”. Mitjançant el desenvolupament del codi basat en el simulador orbital SGP4, s'han generat representacions en un mapa mundi que mostren amb claredat la trajectòria de cadascuna de les naus seleccionades. Permeten observar de manera contínua la seva posició al llarg del període d'estudi.

La segona: “Totes les trajectòries LEO seran molt diferents les unes de les altres, les úniques que tindran la mateixa forma seran les que formen part de la mateixa constel·lació.” s'ha vist que és parcialment correcta. Els satèl·lits que formen part d'una mateixa constel·lació comparteixen alguns elements que els diferencien d'altres. Un d'ells és la forma de la trajectòria que descriuen al cel, però no sempre es compleix. Hi ha satèl·lits que poden manifestar la mateixa forma perquè presenten angles orbitals similars. Per exemple, els que sempre passen pels pols (òrbita polar) semblaran iguals. Els que coincideixin en la inclinació també poden aparèixer idèntics entre ells, per tant, no totes les trajectòries dels satèl·lits LEO seran completament distintes les unes de les altres. El que realment les distingeixen és que al llarg dels dies mantenen la mateixa diferència longitudinal.

La tercera: “És possible determinar els períodes de visibilitat per cada satèl·lit pel dia 1 d'agost, la quantitat de minuts totals, el moment amb la quantitat màxima i l'òptim per dur a terme una observació aficionada.” és en part correcta. És possible determinar els períodes de visibilitat per cada satèl·lit per un dia concret, la quantitat de minuts totals i el moment amb la màxima quantitat de satèl·lits visibles de manera teòrica. Però a la pràctica, pot ser que no s'adeqüi 100% a la realitat. El simulador comprova si està per sobre de l'horitzó respecte a l'institut. En cas d'afirmatiu vol dir que existirà la possibilitat que sigui visible. Si dona el contrari vol dir que serà impossible veure-ho perquè la Terra el bloqueja i queda ocult. És per això que a vegades es veu que quan el satèl·lit està enmig de l'oceà Atlàntic, també es considera visible. No és que estigui malament. Hi ha la possibilitat de veure'l, però es requerirà que no hi hagi obstacles entremig de la línia de visió i no serà tan fàcil d'observar com si el tinguéssim directament a sobre nostre. Així doncs, per realitzar una observació aficionada aquesta informació és útil; tanmateix, s'ha de tenir en compte les variables que podrien influir. L'hora òptima per poder-los observar amb telescopi és després de la posta de sol o poc abans de l'alba, ja que en aquests moments el cel és prou fosc per veure el punt brillant. El satèl·lit encara està il·luminat per la llum solar, per tant, una mica abans de les 4:30 UTC (6:30 CEST) o després de les 19:30 UTC (21:30 CEST). No obstant això, l'1 d'agost només hi havia la possibilitat d'observar 2 satèl·lits dels estudiats durant

aquestes hores. A les 3:04-3:16 UTC (5:04-5:16 CEST) i 20:19-20:28 UTC (22:19-22:28 CEST). En els moments en què tres són visibles de manera teòrica, en realitat, la Terra ja ha presentat la seva ombra i estan completament a les fosques. Això provoca que sigui impossible observar-los. Només es podrien veure amb càmeres d'alta sensibilitat.

La hipòtesi: "A mesura que passa el temps, els fitxers TLE utilitzats perdran precisió, s'aniran quedant obsolets i com a resultat el simulador orbital donarà resultats no exactes. No seran útils per fer prediccions acurades al cap de 5 dies." s'ha mig corroborat. Seguint els resultats obtinguts, si es vol predir la posició exacta del satèl·lit en més de cinc dies, no s'hauria d'utilitzar un mateix fitxer TLE per obtenir resultats amb el simulador orbital SGP4, perquè comença a perdre exactitud. Tot i això, encara es podria emprar per planificar observacions, ja que el satèl·lit encara es veurà dins la zona esperada i l'hora d'aparició només variarà una mica. Per més de 20 dies més tard només tindrà sentit utilitzar-ho per fer estimacions molt àmplies, com per saber si existeix la possibilitat que el satèl·lit sigui visible. La predicció ja deixa de ser útil per planificacions d'observacions serioses. Refutant la hipòtesi, s'ha vist que l'error no augmenta de manera estrictament lineal. En el cas de la latitud i la distància total s'ha observat que presenta una excepció en el resultat de 5 dies més tard. En els posteriors, sí que s'ha vist que ha anat seguit aproximadament un patró. Aleshores, els TLE són fiables per a fer prediccions d'uns 20 dies endavant aproximadament com a màxim. Per fer-ne de més llunyanes, caldrà utilitzar el fitxer d'una data més pròxima del moment que es vol predir, ja que la precisió disminueix amb el temps.

Pel que fa als objectius que han estat necessaris per donar resposta a les hipòtesis s'han pogut complir tots. En primer lloc, s'ha desenvolupat amb èxit un programa basat en el simulador orbital SGP4 que permet representar visualment les trajectòries dels satèl·lits LEO seleccionats en un mapa mundi 2D, mostrant la seva posició i visibilitat en temps futur. S'havia dut a terme una primera versió basada en un bucle que s'encarregava d'anar dibuixant el nou punt i després esborrar-ho va resultar no satisfactòria a causa de la lentitud i incapacitat de fer salts temporals per arribar al dia volgut amb més rapidesa. La segona basada en una animació com a tal, va aconseguir solucionar els problemes topats anteriorment. En segon lloc, s'han processat les dades de visibilitat des del punt de referència de l'Institut Pere Alsius i Torrent, que permet identificar les franges horàries amb major nombre de satèl·lits visibles, així com la durada diària de visibilitat de cadascuna de les naus i determinar el moment ideal de l'1 d'agost del 2025 per dur a terme una observació aficionada. Finalment, s'ha avaluat la validesa del recurs comparant les prediccions amb fitxers TLE actuals, s'han obtingut els errors i s'ha valorat la seva importància.

La dificultat del treball ha tingut lloc en la part pràctica, en la part de la creació del codi, ja que han sorgit errors constantment. Alguns eren estilístics, d'altres eren fruit d'un mal raonament o per falta de consideració. On n'hi ha hagut més, ha estat en la de la creació dels botons de control. Per realitzar salts temporals s'han hagut d'establir límits. Sense aquests, a vegades es feien salts més enllà de les dades que es tenia i com a resultat sorgien trajectòries errònies. D'altra banda, per esborrar trajectòries del dia anterior es va veure que buidar les llistes de les coordenades implicava que a l'hora de fer salts temporals enrere faltessin dades. Per aquesta raó, s'ha ideat la solució de reconstruir tota la trajectòria anterior per desplaçar-se en el temps. L'eliminació de dades no serà un problema aleshores.

Per descomptat que el projecte no és perfecte i hi ha diversos aspectes a millorar.

La primera seria perfeccionar el mapa mundi afegint-li relleu i la funció d'apropar-se i allunyar-se depenent dels detalls que l'usuari desitgi. Certes regions, sobretot les que disposen de superfícies més reduïdes, resulten difícils de visualitzar sense aquesta millora. Complicaria la localització precisa sobre el mapa. Addicionalment, hauria donat el producte final un toc de major professionalitat i el faria més atractiu visualment.

Un altre aspecte seria haver intentat incrementar l'efectivitat dels resultats. En el treball s'ha pogut detectar que es perd precisió a mesura que passen els dies i, per tant, no es pot utilitzar per fer simulacions durant períodes llargs del futur. Davant d'aquesta necessitat potser hauria estat bé haver pogut modificar el simulador d'alguna manera per intentar resoldre-ho, tot i que no era una tasca que entrava dins dels objectius plantejats al principi. Es podria haver creat un codi per automatitzar el procés d'actualitzar els TLE cada vegada que se'n publiquin de nous a la pàgina web, però per l'extensió del treball, falta de temps i coneixement no s'ha dut a terme.

En definitiva, hi ha molts aspectes a millorar, però, així i tot, el treball realitzat ha permès assolir els objectius proposats. A més a més d'haver superat les expectatives que es tenien. Tenint en compte el punt de partida, tot el temps dedicat i on he pogut arribar s'està satisfeta amb els resultats.

Personalment, aquest treball ha anat més enllà de donar resposta a hipòtesis que foren fruit de la curiositat. M'ha permès adquirir una gran diversitat de coneixement dels quals no en sabia res, com conèixer amb profunditat el funcionament dels satèl·lits, els seus sistemes de comunicació, què són els simuladors orbitals i fins i tot a com crear una animació. Sempre m'ha interessat aquest àmbit i m'ha proporcionat una visió global del que és treballar amb les eines que utilitzen els professionals del camp.

En conclusió, el simulador orbital SGP4 respecte a la visibilitat de satèl·lits és una molt bona eina per planificar observacions, però això no vol dir que es pugui prendre com a infal·lible. Quant a la predicció de la posició, durant els primers 5 dies, té una exactitud molt alta, que permet que sigui utilitzada per professionals. Tanmateix, superat aquest llindar es recomana actualitzar els TLE més endavant per mantenir la precisió. Pel que fa a la representació del seu moviment és òptima: mostra com es veurien les trajectòries orbitals en un mapa del món, 2D, no rectes, sinó amb curvatures sinusoidals. No només és de gran ajuda per a investigadors i professionals de l'aeronàutica, també ho són per la societat: facilita la comprensió del moviment dels satèl·lits, ajuda a planificar observacions astronòmiques i contribueix a l'educació i divulgació científica.

10. FONTS CONSULTADES

10.1 Bibliografia

- Meyers, R. A. (2003). *Encyclopedia of Physical Science and Technology* (3rd edition). Academic Press.
- Montenbruck, O., i Gill, E. (2000). *Satellite Orbits*. Springer.
- Russo, C., Sarobe, M., Lencina, P., i Ahmad, T. (2023). *Introducción a la programación con Python*. Marcombo.
- Sebestyen, G., Fujikawa, S., Galassi, N., i Chuchra, A. (2018). *Low Earth Orbit Satellite Design*. Springer.

10.2 Webgrafia

- Agencia Iberoamericana para la Difusión de la Ciencia y la Tecnología. (2009, 29 de juliol). *Lanzado con éxito el Deimos-1, primer satélite español para la observación de la Tierra*. <https://www.dicyt.com/noticias/lanzado-con-exito-el-deimos-1-primer-satelite-espanol-para-la-observacion-de-la-tierra>
- Alén Space. (s.d.). *A Basic Guide to Nanosatellites*. <https://alen.space/basic-guide-nanosatellites/>
- Anywaves. (2023, 22 març). *What is a satellite made of? Components and subsystems*. <https://anywaves.com/resources/blog/what-is-a-satellite-made-of-components-and-subsystems/>
- Betz, E. (2020, 21 juliol). *Apollo-Soyuz Mission: When the Space Race Ended*. Astronomy. <https://www.astronomy.com/space-exploration/apollo-soyuz-mission-when-the-space-race-ended/>
- BlackSky. (2021, 13 agost). *Low Earth Orbit (LEO) satellites*. <https://www.blacksky.com/low-earth-orbit-leo-satellites/>
- Boudrie, J., Shea, E., Pyzdrowski, H., Brisker, K., Fiori, P., Anderson, M., Rausch, J., Mead, P., Kota, K. R., & Lacy, T. E. (2023). *Design and evaluation of additively-manufactured MMOD satellite shielding*. Research Gate. https://www.researchgate.net/publication/367334398_Design_and_Evaluation_of_Additively-Manufactured_MMOD_Satellite_Shielding
- CelesTrak. (s.d.). *Frequently asked questions: Two-line element set format*. <https://celestrak.org/columns/v04n03/>
- CelesTrak. (s.d.). *Two-line element set format*. <https://celestrak.org/NORAD/documentation/tle-fmt.php>
- Christiansen, E., Lear, D., & Hyde, J. (2018). *Micro-meteoroid and orbital debris (MMOD) protection overview*. NASA Johnson Space Center.

- <https://ntrs.nasa.gov/api/citations/20190001193/downloads/20190001193.pdf>
- DHL. (2024, 24 agost). *How LEO satellites will elevate logistics*.
<https://www.dhl.com/global-en/delivered/innovation/how-leo-satellites-will-elevate-logistics.html>
- Dolph Microwave. (2024, 24 d'abril). *What is the feeding structure of an antenna*.
https://www.dolphmicrowave.com/default/what-is-the-feeding-structure-of-an-antenna/#Feed_Array
- Equipo editorial, Etecé (2024, 23 octubre). *Satélites artificiales*. Enciclopedia Humanidades.
<https://humanidades.com/satelites-artificiales/>.
- European Space Agency. (2019, 11 desembre). *Solar Orbiter solar array deployment*.
https://www.esa.int/ESA_Multimedia/Videos/2019/12/Solar_Orbiter_solar_array_deployment
- European Space Agency. (2020, 7 agost). *ESA's 'first satellite' COS-B*.
https://www.esa.int/About_Us/50_years_of_ESA/ESA_s_first_satellite_COS-B
- European Space Agency. (s. d.). *Satellite frequency bands*.
https://www.esa.int/Applications/Connectivity_and_Secure_Communications/Satellite_frequency_bands
- European Space Agency. (s.d.). *The geostationary orbit*.
https://www.esa.int/Education/3_The_geostationary_orbit
- European Space Agency. (s. d.). *Thermal Control*.
https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Thermal_Control
- Hakima, H., Bazzocchi, M. C. F., & Almstrom, B. (2022). *Analysis of satellite drag coefficients based on physical and orbital specifications*. IEEE Xplore.
<https://ieeexplore.ieee.org/document/9843417>
- Kirvan, P. (2023, 21 juny). *What are uplinks and downlinks?*. TechTarget.
<https://www.techtarget.com/searchmobilecomputing/definition/downlink-and-uplink>
- Kruczek, T. (2023, 7 octubre). *Demystifying the USSPACECOM Two-Line Element Set Format*. KeepTrack.Space.
<https://keeptrack.space/deep-dive/two-line-element-set/>
- Maury Microwave. (2025, 25 març). *How Do Natural Effects & RF Impairments Impact Satellite Transmissions*.
<https://blog.maurymw.com/how-do-natural-effects-rf-impairments-impact-satellite-transmissions>
- NASA Goddard Space Flight Center. (2021, 27 de setembre). *Landsat 9*.
<https://landsat.gsfc.nasa.gov/satellites/landsat-9/>
- NASA Jet Propulsion Laboratory. (s.d.). *Attitude and orbit control system*.

- <https://gracefo.jpl.nasa.gov/attitude-and-orbit-control-system/>
- New Mexico Museum of Space History. (2021). *Konstantin E. Tsiolkovsky*.
<https://nmspacemuseum.org/inductee/konstantin-e-tsiolkovsky/>
- Pultarova, T. (2025, 1 agost). *Starlink satellites: Facts, tracking and impact on astronomy*.
 Space.com. <https://www.space.com/spacex-starlink-satellites.html>
- Quesada Pereira, F. D. (2010, 15 d'octubre). *Tema 3: Comunicaciones móviles*. Universidad Politécnica de Cartagena.
https://ocw.bib.upct.es/pluginfile.php/6091/mod_resource/content/1/tema3_2009.pdf
- Qu, Z., Zhang, G., Cao, H., & Xie, J. (2017). *LEO satellite constellation for Internet of Things*. IEEE Xplore.
<https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8002583>
- RF Wireless World. (s.d.). *Satellite Ground Station Basics*.
<https://www.rfwireless-world.com/articles/satellite-ground-station-basics>
- Rodzi, D. (2022, 13 maig). *History of satellites*. Soar.
<https://about.soar.earth/blog-pages/history-of-satellites>
- Satellite. (2025, 4 juny). En *Wikipedia*. <https://en.wikipedia.org/wiki/Satellite>
- Science Learning Hub, University of Waikato. (2013, 27 març; actualitzat el 2022, 10 agost). *History of satellites-timeline*.
<https://www.sciencelearn.org.nz/resources/1905-history-of-satellites-timeline>
- Sergieieva, K. (2023, 22 març). *Types of satellites by orbits, functions, and practical uses*. EOS Data Analytics. <https://eos.com/blog/types-of-satellites/>
- Sodders, L. (2023, 18 juliol). *LEO, MEO or GEO? Diversifying orbits is not a one-size-fits-all mission*. Space Operations Command.
<https://www.spoc.spaceforce.mil/News/Article-Display/Article/3462529/leo-meo-or-geo-diversifying-orbits-is-not-a-one-size-fits-all-mission-part-1-of>
- Starlink. (s.d.). *Starlink technology*.
https://www.starlink.com/technology?srsId=AfmBOopOLZMc3G1-N9rFa2CtNZreljKkWQeFLZ1EbYq5_gJqP5bfBrm
- Zhang, R., Yang, K., Zhang, J., Bi, S., & Lacy, T. E. (2024). *Overview and key technology of the membrane drag sail for low Earth orbit satellite deorbit*. Sciopen.
<https://www.sciopen.com/article/10.34133/space.0115>

11. ÍNDEX D'IMATGES

Fig. 1 Sputnik-1.....	9
Fig. 2 Parts d'una el·lipse.....	12
Fig. 3 Esfera terrestre amb les coordenades equatorials, altitud geoestacionària.....	15
Fig. 4 Esfera terrestre amb les coordenades equatorials, altitud mitjana.....	16
Fig. 5 Esfera terrestre amb les coordenades equatorials, altitud baixa.....	17
Fig. 6 Esquema d'un satèl·lit.....	17
Fig. 7 Sistema de comunicació típic.....	21
Fig. 8 Diferents parts d'un fitxer TLE.....	28
Fig. 9 Sistema de catalogació Alpha-5.....	29
Fig. 10 Esquema del programa.....	33
Fig. 11 Fitxers del simulador.....	33
Fig. 12 Importacions de llibreries.....	34
Fig. 13 Transformació de longitud i latitud en radians i obtenció de coordenades x, y, z.....	35
Fig. 14 Fitxers oberts preparats per ser llegits i per escriure-hi.....	35
Fig. 15 Variable <i>visible</i>	35
Fig. 16 Variable <i>prn</i>	35
Fig. 17 Càlcul de l'angle d'azimuth i adaptació del resultat a les nostres necessitats.....	36
Fig. 18 Institut Pere Alsius i Torrent a Google Earth.....	38
Fig. 19 Coordenades de l'institut Pere Alsius i Torrent des de Google Earth.....	38
Fig. 20 Modificant les coordenades per les obtingudes.....	39
Fig. 21 Modificació de la línia de codi per obtenir el text imprès actualitzat.....	39
Fig. 22 Comprovació del text imprès.....	39
Fig. 23 Fitxers TLE dels satèl·lits LEO que se simularan.....	40
Fig. 24 Canvis realitzats.....	41
Fig. 25 Activació de l'entorn virtual "venv".....	43
Fig. 26 Descarregant llibreries.....	43
Fig. 27 Reajust automàtic si no fas servir <code>ax.set_extent</code>	45
Fig. 28 Resultat erroni que prové de prémer el botó "anterior satèl·lit" en el primer.....	60
Fig. 29 Moment de la simulació seleccionada.....	61
Fig. 30 Comprovació de les coordenades del mapa. Posició de LandSat-9 el 25/7/2025 a les 00:16 en el mapa de l'animació.....	61
Fig. 31 Comprovació de les coordenades del mapa. Posició de LandSat-9 en Google Maps.....	62
Fig. 32 Predicció de la trajectòria de LandSat-9 (49260) l'1 d'agost de 2025.....	63

Fig. 33 Gràfica visibilitat (1=visible, 0=invisible)-temps (h) de LandSat-9 (49260) l'1 d'agost de 2025.....	63
Fig. 34 Predicció de la trajectòria de LandSat-8 (39084) l'1 d'agost de 2025.....	64
Fig. 35 Gràfica visibilitat (1=visible, 0=invisible)-temps (h) de LandSat-8 (39084) l'1 d'agost de 2025.....	64
Fig. 36 Posició de LandSat-9 (49260) el 2 d'agost de 2025 a les 3:57.....	65
Fig. 37 Posició LandSat-8 (39084) el 2 d'agost de 2025 a les 3:57.....	65
Fig. 38 Posició Terra (25994) el 2 d'agost de 2025 a les 3:57.....	66
Fig. 39 Posició de LandSat-9 (49260) el 9 d'agost de 2025 a les 2:57.....	66
Fig. 40 Posició LandSat-8 (39084) el 9 d'agost de 2025 a les 2:57.....	66
Fig. 41 Posició Terra (25994) el 9 d'agost de 2025 a les 2:57.....	67
Fig. 42 Predicció de la trajectòria de Terra (25994) l'1 d'agost de 2025.....	67
Fig. 43 Gràfica visibilitat (1=visible, 0=invisible)-temps (h) de Terra (25994) l'1 d'agost de 2025.....	67
Fig. 44 Posició de Deimos-1 (35681) el 25 de juliol de 2025 a les 6:10.....	68
Fig. 45 Posició de Deimos-2 (40013) el 25 de juliol de 2025 a les 6:10.....	68
Fig. 46 Predicció de la trajectòria de Deimos-1 (35681) l'1 d'agost de 2025.....	69
Fig. 47 Gràfica visibilitat (1=visible, 0=invisible)-temps (h) de Deimos-1 (35681) l'1 d'agost de 2025.....	69
Fig. 48 Predicció de la trajectòria de Deimos-2 (40013) l'1 d'agost de 2025.....	69
Fig. 49 Gràfica visibilitat (1=visible, 0=invisible)-temps (h) de Deimos-2 (40013) l'1 d'agost de 2025.....	70
Fig. 50 Predicció de la trajectòria de Starlink-1010 (44716) l'1 d'agost de 2025.....	70
Fig. 51 Gràfica visibilitat (1=visible, 0=invisible)-temps (h) de Starlink-1010 (44716) l'1 d'agost de 2025.....	71
Fig. 52 Predicció de la trajectòria de Oneweb-0012 (44057) l'1 d'agost de 2025.....	71
Fig. 53 Gràfica visibilitat (1=visible, 0=invisible)-temps (h) de Oneweb-0012 (44057) l'1 d'agost de 2025.....	72
Fig. 54 Gràfica quantitat de satèl·lits visibles-temps (h) de l'1 d'agost de 2025.....	72
Fig. 55 Modificació de paràmetres. La data d'iniciació per 30/07/2025 i la duració per 1 dia.....	73
Fig. 56 Introducció de TLE de LandSat-9 del 30 de juliol a les 11:31.....	73
Fig. 57 Línia del fitxer de sortida amb les coordenades de posició per LandSat-9 el 30 de juliol a les 11:31.....	73
Fig. 58 Visualització des de l'animació creada la posició prevista de LandSat-9 del 30 de juliol a les 11:31.....	74

Fig. 59 Visualització des de Google Maps la posició real de LandSat-9 del 30 de juliol a les 11:31.....	75
Fig. 60 Modificació de paràmetres. La data d'iniciació és canviada per 04/08/2025.....	75
Fig. 61 Introducció de TLE de LandSat-9 del 4 d'agost a les 5:14.....	75
Fig. 62 Línia del fitxer de sortida amb les coordenades de posició per LandSat-9 el 4 d'agost a les 5:14.....	75
Fig. 63 Visualització des de l'animació creada la posició prevista de LandSat-9 del 4 d'agost a les 5:14.....	77
Fig. 64 Visualització des de Google Maps la posició real de LandSat-9 del 4 d'agost a les 5:14.....	77
Fig. 65 Modificació de paràmetres. La data d'iniciació és canviada per 09/08/2025.....	77
Fig. 66 Introducció de TLE de LandSat-9 del 9 d'agost a les 5:14	77
Fig. 67 Línia del fitxer de sortida amb les coordenades de posició per LandSat-9 el 8 d'agost a les 10:30.....	78
Fig. 68 Visualització des de l'animació creada la posició prevista de LandSat-9 del 8 d'agost a les 10:30.....	79
Fig. 69 Visualització des de Google Maps la posició real de LandSat-9 del 8 d'agost a les 10:30.....	79
Fig. 70 Modificació de paràmetres. La data d'iniciació és canviada per 14/08/2025.....	80
Fig. 71 Introducció de TLE de LandSat-9 del 14 d'agost a les 10:48.....	80
Fig. 72 Línia del fitxer de sortida amb les coordenades de posició per LandSat-9 el 14 d'agost a les 10:48.....	80
Fig. 73 Visualització des de l'animació creada la posició prevista de LandSat-9 del 14 d'agost a les 10:48.....	81
Fig. 74 Visualització des de Google Maps la posició real de LandSat-9 del 14 d'agost a les 10:48.....	81
Fig. 75 Modificació de paràmetres. La data d'iniciació és canviada per 19/08/2025.....	82
Fig. 76 Introducció de TLE de LandSat-9 del 19 d'agost a les 4:31.....	82
Fig. 77 Línia del fitxer de sortida amb les coordenades de posició per LandSat-9 el 19 d'agost a les 4:31.....	82
Fig. 78 Visualització des de l'animació creada la posició prevista de LandSat-9 del 19 d'agost a les 4:31.....	83
Fig. 79 Visualització des de Google Maps la posició real de LandSat-9 del 19 d'agost a les 4:31.....	83
Fig. 80 Modificació de paràmetres. La data d'iniciació és canviada per 24/08/2025.....	84
Fig. 81 Introducció de TLE de LandSat-9 del 24 d'agost a les 14:43.....	84

Fig. 82 Línia del fitxer de sortida amb les coordenades de posició per LandSat-9 el 24 d'agost a les 14:43.....	84
Fig. 83 Visualització des de l'animació creada la posició prevista de LandSat-9 del 24 d'agost a les 14:43.....	85
Fig. 84 Visualització des de Google Maps la posició real de LandSat-9 del 24 d'agost a les 14:43.....	85
Fig. 85 Simulador SGP4 (fitxer comprimit).....	99
Fig. 86 Simulador amb tots els codis (fitxer comprimit).....	99
Fig. 87 Fitxers TLE.....	99
Fig. 88 main.py original (PDF).....	99
Fig. 89 main.py adaptat (PDF).....	99
Fig. 90 1a versió codi (PDF).....	100
Fig. 91 2a versió codi (PDF).....	100
Fig. 92 Gràfics.....	100
Fig. 93 Mostra resultats (vídeo).....	100

12. ANNEXOS

12.1 Annex 1: Simulador SGP4 proporcionat pel programa Argó de la Universitat Autònoma de Barcelona (fitxer comprimit)



Fig. 85 *Simulador SGP4 (fitxer comprimit)*
Font: pròpia

12.2 Annex 2: Simulador adaptat amb la 1a i 2a versió del codi per visualitzar la trajectòria (fitxer comprimit)



Fig. 86 *Simulador amb tots els codis (fitxer comprimit)*
Font: pròpia

12.3 Annex 3: Fitxers TLE dels satèl·lits simulats (PDF)



Fig. 87 *Fitxers TLE*
Font: pròpia

12.4 Annex 4: main.py (fitxer principal) original (PDF)



Fig. 88 *main.py original (PDF)*
Font: pròpia

12.5 Annex 5: main.py (fitxer principal) adaptat (PDF)



Fig. 89 *main.py adaptat (PDF)*
Font: pròpia

12.6 Annex 6: 1a versió del codi per visualitzar la trajectòria (PDF)



Fig. 90 1a versió
codi (PDF)
Font: pròpia

12.7 Annex 7: 2a versió del codi per visualitzar la trajectòria (PDF)



Fig. 91 2a versió
codi (PDF)
Font: pròpia

12.8 Annex 8: Gràfics



Fig. 92 Gràfics
Font: pròpia

12.9 Annex 9: Mostra dels resultats (vídeo).



Fig. 93 Mostra
resultats (vídeo)
Font: pròpia