

Ejercicio 1a) Completar la función en MATLAB : `function df=get_df(f)`
`...`
`return`

para que devuelva un **vector fila df** de tamaño 1xN con las diferencias finitas a partir de los datos de entrada (vector f, también 1xN).

Probad la función con un vector de los ocho primeros valores de la coordenada Z del satélite con PRN=3 guardados en el fichero 'IGS12651.sp3'. Para ello leer los datos (usando read_sp3) en una estructura sp y hacer:

`Z = sp.XYZ(3,1:8,find(sp.prn==3));` Comprobad que estos valores son:

`+8396986.531 +5778566.750 +3061782.186 +292951.342 -2480860.984 ...`

Aplicad vuestra función `dZ=get_df(Z)` y usar `fprintf('%0.3f ',dZ)` para ver los resultados con 3 decimales. Debéis obtener **exactamente**:

`8396986.531 -2618419.781 -98364.783 46318.503 746.295 -771.038 7.614 10.240`

Repetir el cálculo para la coordenada X y volcad vuestros resultados..

Adjuntar código de vuestra función (se valorará no usar más memoria que el vector de datos de salida df, donde se ha copiado el vector de datos original).

Ejercicio 1b): Completar la función `get_S()` para que reciba un valor de s y devuelva el vector de coeficientes $S(k)$ que acompañan a las diferencias finitas en el polinomio de interpolación. Para calcularlas usamos la fórmula recursiva vista en clase, que modificaremos para tener en cuenta que MATLAB empieza a contar los índices desde 1:

$$S_1 = 1, \quad S_{k+1} = S_k \frac{(s-k+1)}{k}, \quad \text{con } k = 1, \dots, N-1$$

donde N el número de nodos usados en la interpolación. Devolver el vector S como un **vector columna N x 1**.

Los valores obtenidos para los $S(k)$ deberían ser: $1, s, \frac{s(s-1)}{2}, \frac{s(s-1)(s-2)}{3}, \dots$, etc.

Como ejemplo, la función `get_S` debe daros los siguientes resultados:

`s=3.5,N=8: 1.0000 3.5000 4.3750 2.1875 0.2734 -0.0273 0.0068 -0.0024`

`s=2.5,N=6: 1.0000 2.5000 1.8750 0.3125 -0.0391 0.0117`

Dar los coeficientes obtenidos (con 5 decimales) para `s=3.1, N=8` y `s=2.9, N=6`

Adjuntar código de vuestra función.

Ejercicio 2: el fichero 'IGS12651.SP3' corresponde al segundo (0,1,2,...,6) día (lunes) de la semana GPS 1265. El tiempo del primer dato `sp.tow(1)`, es de 86400 segundos, que corresponde al número de segundos en el día que ya ha pasado de la semana (el domingo, el primer día de la semana GPS). Queremos interpolar la posición de un satélite a las 9h 20m de ese lunes.

a) ¿Cuántos segundos han transcurrido del lunes? ¿A que `tow` (segundos dentro de la semana) corresponde esa hora? ¿Entre qué dos nodos del fichero cae ese tiempo? Comprobarlo volcando los valores de `sp.tow` para esos nodos. Determinar rango de índices de los que extraeríamos los datos para la tabla de interpolación suponiendo que usamos $N=8$ nodos en la interpolación.

b) Extraed (del campo `sp.XYZ`) los N datos de la coordenada X en esos nodos para el satélite con $PRN=25$. Calculad sus diferencias finitas dX (usando la función del ejercicio 1a).

c) A partir del tiempo t y el espaciado entre nodos h determinar el valor de s (ver indicaciones en las transparencias). A partir de s , calcular el vector de los coeficientes S_k (para $N=8$) con la función del ejercicio 1b).

e) Combinad las diferencias finitas con los coeficientes S_k para calcular el valor interpolado de X . Como el vector de diferencias dX es un vector fila ($1 \times N$) y el vector de coeficientes S un vector columna ($N \times 1$), nos basta hacer el producto escalar de ambos: $X_{int} = dX * S$:

$$X_{int} = \sum_k dX_k S_k = \langle d\bar{X}, \bar{S} \rangle$$

Usad la siguiente traza para verificar que vuestro código va calculando de forma correcta todos los pasos. [Adjuntar código de vuestro script](#)

Tiempo $t=120000$. Cae entre nodos 38 y 39

Índices de los nodos ($N=8$) a usar: desde 35 a 42

**Valores X : 23899566.29 23141958.06 22202785.17 21120835.89
19937535.26 18695306.88 17435890.64 16198687.97**

dif X : 23899566.29 -757608.23 -181564.66 38788.26 2636.80 -1638.28 -43.91 71.30

Valor de s : 3.333 Coefs $\{S_k\}=1.000 3.333 3.889 1.728 0.144 -0.019 0.005 -0.002$

Valor interpolado $X_{int}=20735573.05$ (mt)

Si se desea repetir la interpolación (en el mismo instante de tiempo) para las otras coordenadas Y , Z o el error de reloj dT , ¿es necesario volver a calcular los coeficientes S_k ?

Si ahora queremos interpolar X a las 9h 25m, ¿es necesario volver a calcular el vector de diferencias finitas? ¿Qué es lo que cambia en este nuevo caso?

¿Valor de la interpolación de X en la nueva hora (9h 25m)?

Ejercicio 3: Escribir una función **interp_sat.m**:

```
function [XYZ, cdT]=interp_sat(sp,tow,PRN,N)
% Parametros entrada: sp -> estructura con datos pertinentes.
%                     tow -> tiempo de la semana en seg
%                     PRN del sat en el que estamos interesados
%                     N, número de nodos a usar.
% Parametros SALIDA: XYZ=vector 3x1 con la posición interpolada (en mt)
%                  cdT = error reloj (en mt) interpolado en tiempo t.
```

Los parámetros de entrada son una estructura *sp* con los datos satelitales de todo un día, un tiempo GPS *tow*, un PRN y el número de nodos *N* a usar en la interpolación (*N* debe ser par). La función debe devolver la interpolación de la posición XYZ (vector 3x1) y el error de reloj (*cdT*) del satélite en el *tow* dado. Asumiremos que el tiempo pedido es tal que todos los nodos necesarios se encuentran dentro de la estructura *sp*.

Nos basaremos en el código del ejercicio anterior, encapsulando el script que ya tenemos en una función y completando los siguientes pasos:

- Calcular el valor de *s* y el vector de coeficientes *S* (*N*x1) a partir de *tow*. Determinar el rango de nodos a usar y el índice del satélite PRN en las tablas. Esto es idéntico a lo que ya tenáis hecho en el ejercicio anterior.
- Reservar matriz **dif_XYZ** de tamaño 3x*N*. Con un bucle desde *k*=1 a 3, extraer las correspondientes coordenadas *X,Y,Z*, calcular sus diferencias finitas y guardar los resultados en las sucesivas filas de *dif_XYZ*.
- Multiplicando la matriz *dif_XYZ* (3 x *N*) por el vector *S* (*N* x 1) se obtiene un vector 3 x 1 con la posición interpolada en las 3 coordenadas (XYZ).
- Finalmente extraer los datos del error de reloj y calcular su tabla de diferencias finitas (vector 1 x *N*). Multiplicarla por el vector *N* x 1 de los coeficientes *S* para obtener un escalar, el error de reloj interpolado *cdT*.

Con los datos de IGS12651.SP3 y usando *N*=8 nodos en la interpolación, la posición XYZ y error de reloj *cdT* (ambos en mt) del satélite con PRN=25 en el tiempo *tow*=120000 son:

```
XYZ = [20735573.05 -4458154.90 -15607934.87]
cdT = 19133.33
```

Usar `fprintf('XYZ= [%2f %2f %2f] cdT = %2f\n',XYZ,cdT)` para volcar los resultados de vuestra función con 2 decimales y verificarlos con los resultados anteriores.

Tras verificar vuestra función, usadla para determinar la posición XYZ + error de reloj *cdT* del satélite con PRN=13 en el tiempo *tow*=130000.

Adjuntar los resultados (con 2 decimales) para el caso de *N*=8 y *N*=6 nodos.

Adjuntar el código de vuestra función.

Ejercicio 4: Ejecutar la orden: `>> sp = read_sp3('IGS12651.SP3');` que lee los datos de las posiciones de los satélites durante 1 día (5 abril 2004) y almacena sus datos en una estructura con los siguientes campos:

- **sp.nsats:** número de satélites de los que hay datos en el fichero. Ese día había 28 satélites operativos por lo que `sp.nsats=28`
- **sp.prn:** vector (1 x nsats) con los números (PRNs) de los satélites.
- **sp.week:** semana GPS (1265 en este caso).
- **sp.delta:** espaciado entre instantes de tiempo (900 sec = 15 min)
- **sp.tow:** vector (1 x 96) con los tiempos en que se "tomaron" los datos ($24 \times 3600 / 900 = 96$, hay 96 "cuartos de hora" en un día).
- **sp.XYZ:** matriz de 3 x 96 x 28 con los datos de las posiciones XYZ (en metros) en los 96 instantes de tiempo de los nsats=28 satélites. Las unidades para las posiciones XYZ son metros
- **sp.dT:** matriz 28 x 96 con los errores de reloj (también en metros) de los nsats=28 satélites disponibles en los 96 instantes de tiempo.

Usando el indexado de MATLAB extraer la siguiente información:

- Hallar en que índices se encuentra la información sobre los satélites con PRN = 15 y PRN = 23 (usar la función `find`). [Adjuntar código y resultado](#)
- Extraer las posiciones X,Y,Z del satélite PRN=15 a lo largo del día como una matriz 3x96. Superponerlas (diferentes colores) en una gráfica en función del tiempo en horas (respecto al inicio del día). [Adjuntar gráfica.](#)
- Hacer un plot 3D de la órbita del satélite anterior usando `plot3(x,y,z)`. ¿Tiene la órbita el aspecto esperado (una elipse)? ¿Por qué puede ser? [Adjuntar gráfica.](#) [Responder](#)
- Extraer el error de reloj del satélite PRN=15 a lo largo de todo el día. Convertir dicho error de metros a microsegundos ($1 \text{ seg} = 10^6 \mu\text{sec}$) y mostrad su gráfica frente al tiempo (en horas). [Adjuntad gráfica.](#)
- La deriva (adelanto o retraso) de un reloj es la variación de su error con el tiempo. [¿Cuál es la deriva del reloj del satélite PRN=15 en \$\mu\text{sec}/\text{día}\$?](#) [¿Y la de los satélites con PRN=21 y PRN=27?](#)

La deriva se suele expresar en partes por millón (ppm). Una deriva de 1 ppm indica que el reloj adelanta o atrasa 1 microsegundo por segundo. Sabiendo que 1 día = 86400 segundos, [¿cuál es la deriva del sat con PRN=15 expresada en ppm?](#) (como comparación, la precisión típica de un reloj de cuarzo es del orden de 1-10 ppm).

Ejercicio 5: De forma similar a como se ha interpolado la posición (XYZ) y error de reloj (cdT) de un satélite podemos interpolar su vector velocidad (Vxyz) y la deriva D de su reloj (lo que el reloj adelanta o atrasa). Para ello, nos basaremos en el hecho de que la velocidad Vxyz y deriva D son las derivadas respectivamente de la posición XYZ y error cdT.

Recordemos que la expresión del polinomio interpolador (para cualquiera de las variables X,Y,Z, dT) era:

$$p(s) = \sum_{k=0}^{N-1} d_k \cdot S_k$$
 donde las $\{S_k\}$ dependen de la variable s y las $\{d_k\}$ son las diferencias divididas para la variable en cuestión.

Si derivamos la expresión anterior con respecto al tiempo obtendremos las derivadas del polinomio interpolador:

$$\frac{dp}{dt} = \frac{dp}{ds} \frac{ds}{dt} = p'(s) \frac{1}{h} = \frac{1}{h} \left(\sum_{k=0}^{N-1} d_k \frac{dS_k}{ds} \right) = \frac{1}{h} \left(\sum_{k=0}^{N-1} d_k \cdot dS_k \right)$$

Salvo el factor 1/h, tenemos una expresión similar a la de la interpolación de la posición: las mismas diferencias finitas de antes $\{d(k)\}$ se mezclan con unos nuevos coeficientes "derivadas" $dS(k)$. Para hallar el valor de estos nuevos coeficientes usaremos una regla recursiva como hicimos para calcular los $\{S_k\}$:

$$dS_1 = 0, \quad dS_{k+1} = \frac{S_k + (s - k + 1) \cdot dS_k}{k}.$$

Empezamos con $dS(1)=0$, e iremos construyendo los sucesivos coeficientes ($k=2, 3, \dots, N$) apoyándonos en los valores anteriores tanto de S como de dS.

Como para calcular los nuevos coeficientes $\{dS\}$ se necesita un bucle muy similar al usado para calcular los coeficientes originales $\{S\}$ lo más sencillo es reusar la función get_S para que calcule ambos coeficientes al mismo tiempo:

```
function [S,dS] = get_S(s,N)
```

La parte de las diferencias divididas no cambia. Si dif_XYZ era la matriz 3 x N con las diferencias finitas de antes, para obtener la posición hacíamos:

dif_XYZ * S → obtenemos vector 3 x 1 con posición XYZ

Si hemos calculado también dS y hacemos:

(dif_XYZ * dS)/h → vector 3 x 1 con las velocidades (Vx,Vy,Vz) en m/s

Repitiendo con la tabla de errores de reloj (acordaros también de la división por h) obtendremos la deriva del reloj en unidades de velocidad (m/s).

Se trata de reescribir la función `get_S` (para que calcule también los nuevos coeficientes `dS`) y ampliar la función `interp_sat()` para que devuelva el vector velocidad `Vxyz` y la deriva `D` como argumentos de salida adicionales:

```
function [XYZ, cdT, Vxyz, D] =interp_sat(sp,t,PRN,N)
% XYZ : vector 3 x 1 con la posición XYZ (m) en tiempo t
% cdT: error de reloj del sat (m) en el tiempo t.
% Vxyz : vector 3 x 1 con la velocidad (Vx,Vy,Vz) del sat (m/s)
% D : deriva del reloj (m/s) en tiempo t.
```

Para `IGS12651.sp3`, `PRN 25`, `t=120000` los valores obtenidos para `dS` son:

```
ds{k} = 0.0000  1.0000  2.8333  2.5556  0.6451  -0.0572  0.0127  -0.0041
```

```
lo que da un resultado (en m/s) de: Vxyz = [-1301.25 1608.45 -2139.06]
                                     D = deriva = 2.48e-04
```

Adjuntar la función `get_S` ampliada (para calcular `dS`) y el código añadido al final de `interp_sat` para calcular `Vxyz` y `D`.

Dar los resultados (coeficientes `dS`, velocidades y deriva `D`) obtenidos para el satélite con `PRN 13` en el instante de tiempo `tow=130000` usando `N=8` nodos.

Ejercicio 6: Una decisión importante es decidir cuantos nodos se usaran en la interpolación, esto es, qué valor dar al parámetro `N` que en nuestra función `interp_sat()` hemos dejado como un valor a especificar.

Se trata de que para los datos de '`IGS12651.SP3`' calculéis los resultados de vuestra interpolación para el satélite `PRN=21` entre las 10h y las 20 horas, con un intervalo de 60 segundos (en total, $10 \times 3600 / 60 = 600$ llamadas a vuestra función). Usar `N=10` nodos de interpolación, esto es, 5 a cada lado. Supondremos que esta es la "verdadera" órbita del satélite.

Repetir el proceso anterior para `N = 2, 4, 6 y 8`.

Usar la coordenada `Z` para comparar y hacer cuatro gráficas de su diferencia entre su interpolación con usando `N = 2, 4, 6, 8` y la "verdad" (`N=10`).

Adjuntar código y gráficas.

A partir de las gráficas, sabiendo que en el uso normal de GPS (debido a la acumulación de otros errores) no esperamos una precisión por debajo de unos 5-10 metros, ¿cuál sería el número adecuado de nodos a usar?

Repetir pero ahora con la variable error de reloj. Pintar de nuevo los errores expresados en metros.

Adjuntar gráfica y comentar los nuevos resultados.