

3.22 The vapor pressure P of water (in bars) as a function of temperature $T[^\circ\text{C}]$ is

T	0	10	20	30	40	50
$P(T)$	0.006107	0.012277	0.023378	0.042433	0.073774	0.12338
T	60	70	80	90	100	
$P(T)$	0.19924	0.31166	0.47364	0.70112	1.01325	

Interpolate these data with the cubic spline $S(x)$. It is also known that $P(5) = 0.008721$, $P(45) = 0.095848$ and $P(95) = 0.84528$. How well does $S(x)$ do at these points?

Se introdujeron a Matlab los valores de la temperatura y la presión para ser interpolados utilizando `spline` y `pchip`, estos resultados se graficaron en el intervalo $[0,100]$ y se muestran en la Fig. 1.

Luego, se introdujeron los puntos $\{(5,0.008721), (45,0.095848), (95,0.84528)\}$ y se calculó el error absoluto con los puntos interpolados por `spline` y `pchip` correspondientes. Los resultados se muestran en la Tabla 1.

Tabla 1. Errores absolutos para $P(5)$, $P(45)$, $P(95)$ con distintas interpolaciones.

$S(x)$	$ P(5) - S(5) $	$ P(45) - S(45) $	$ P(95) - S(95) $
spline	2.7178×10^{-5}	8.4562×10^{-6}	8.3675×10^{-5}
pchip	5.7385×10^{-5}	3.2335×10^{-5}	4.938×10^{-4}

Podemos notar que, aunque de primera vista las interpolaciones se ven iguales (Fig. 1) en realidad difieren en cada punto no interpolado de manera considerable. Se puede concluir que `spline` interpoló mejor los datos que `pchip` ya que todos sus errores son menores en los puntos considerados, de hecho, sus diferencias se incrementan en tanto se incrementa la temperatura.

Para evidenciar mejor este comportamiento, se muestra en la Fig. 2 el punto $(45, P(45))$ en contraste con cada interpolación, donde se puede ver claramente la diferencia entre `pchip` y `spline`.

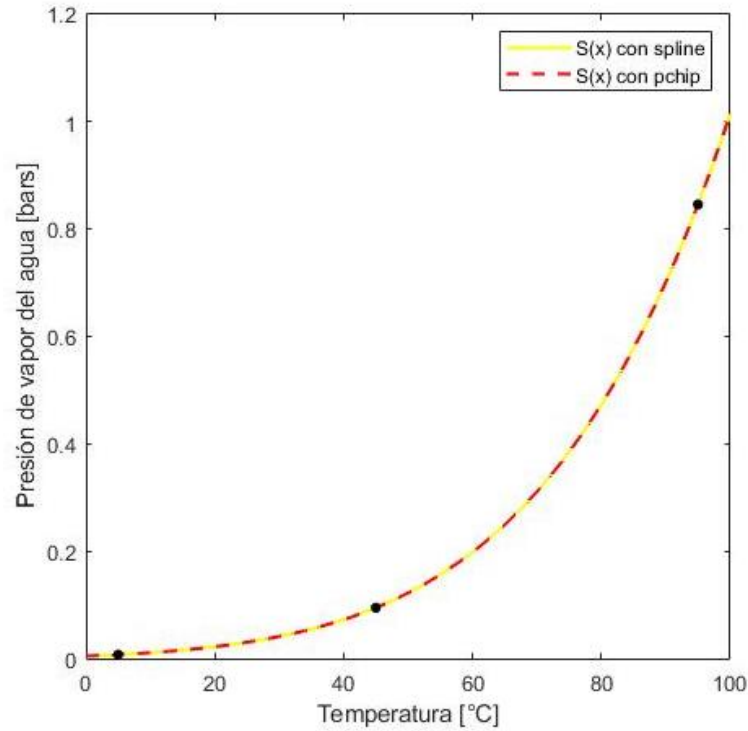


Fig. 1. Interpolaciones de valores de presión de vapor del agua en función de la temperatura con pchip y spline. Se muestran los puntos $\{(5, P(5)), (45, P(45)), (95, P(95))\}$.

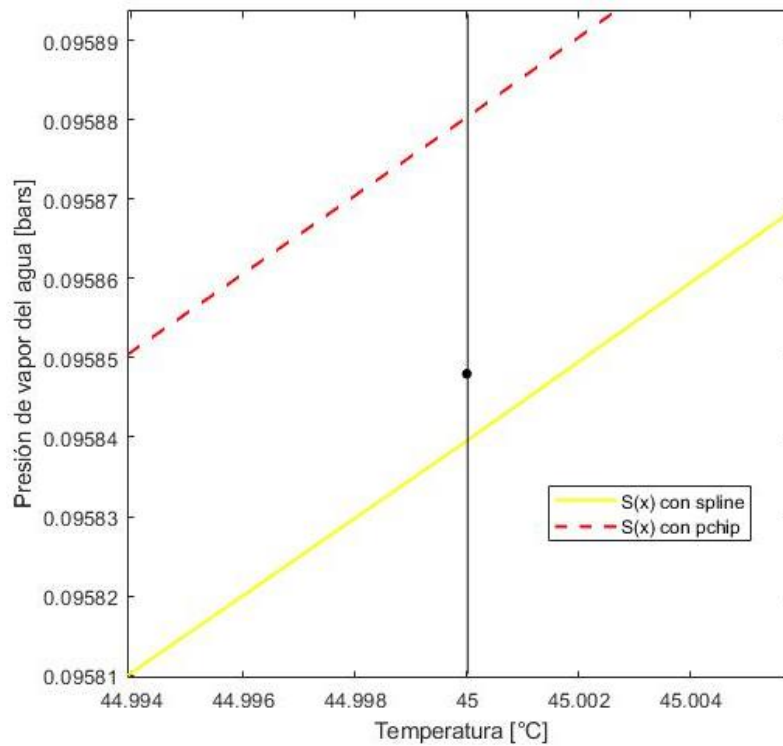


Fig. 2. Detalle del punto $(45, P(45))$. Se aprecia la diferencia entre los valores propuestos por cada método de interpolación.

```
%Introducimos los valores de temperatura y presión
temp=[0:10:100]';
pres=[0.006107,0.012277,0.023378,0.042433,0.073774,0.12338,...
      0.19924,0.31166,0.47364,0.70112,1.01325]';
%Interpolamos con splines cúbicos utilizando "spline" y "pchip"
%Estas funciones regresan una estructura con los polinomios a trozos
pp_spline=spline(temp,pres);
pp_pchip=pchip(temp,pres);
%Graficamos la interpolación de los datos en [0,100]
%Las estructuras anteriores se evalúan con "ppval"
x=[0:0.5:100];
plot(x,ppval(pp_spline,x),'y-','linewidth',1.5); hold on;
plot(x,ppval(pp_pchip,x),'r--','linewidth',1.5); hold on;
legend('S(x) con spline','S(x) con pchip'); axis square;
%Por último insertamos y graficamos los puntos pedidos
x=[5,45,95]; y=[0.008721,0.095848,0.84528];
plot(x(1),y(1),'k.','MarkerSize',14,'handlevisibility','off');
plot(x(2),y(2),'k.','MarkerSize',14,'handlevisibility','off');
plot(x(3),y(3),'k.','MarkerSize',14,'handlevisibility','off');
y_spline=ppval(pp_spline,x); y_spline=abs(y_spline-y);
disp('Errores absolutos de P(5), P(45), P(95) por Spline');
disp(y_spline);
y_pchip=ppval(pp_pchip,x); y_pchip=abs(y_pchip-y);
disp('Errores absolutos de P(5), P(45), P(95) por Pchip');
disp(y_pchip);
```

Command Window

```
>> ejercicio_3_22
Errores absolutos de P(5), P(45), P(95) por Spline
    2.7178e-05    8.4562e-06    8.3675e-05

Errores absolutos de P(5), P(45), P(95) por Pchip
    5.7385e-05    3.2335e-05    0.0004938
```

3.23 The following data give the absorbance of light (A) as a function of wavelength (λ) for vanadyl D-tartrate dimer.

λ	>3125	>3250	3375	>3500
$A(\lambda)$	0.700	0.572	0.400	0.382
λ	3625	>3750	3875	>4000
$A(\lambda)$	0.449	0.560	0.769	0.836
λ	4125	>4250	4325	>4500
$A(\lambda)$	0.750	0.530	0.315	0.170
λ	4625	>4750	4875	>5000
$A(\lambda)$	0.144	0.183	0.252	0.350

Use the cubic spline $S(x)$ to interpolate the nine indicated ($>$) data points. Explore the effects of scaling and shifting the independent variable ($x = \lambda$) with each of the following.

- The data as is.
- Replace x by $\frac{x}{1000}$ for all inputs.
- Replace x by $\frac{x-4000}{1000}$ for all inputs

For each case evaluate $S(x)$ at the remaining noninterpolated wavelengths. How well do these values compare with the known absorbances? Does shifting and/or scaling affect the accuracy of $S(x)$?

Primero se interpolaron los nueve datos seleccionados con `spline` y `pchip` y se graficaron las funciones interpolantes obtenidas en el intervalo $[3125, 5000]$ (Fig. 3). Se obtuvieron las características oscilaciones del método de splines cúbicos entre cada par de nodos, mientras que para `pchip` se obtuvo una curva menos suave sin oscilaciones.

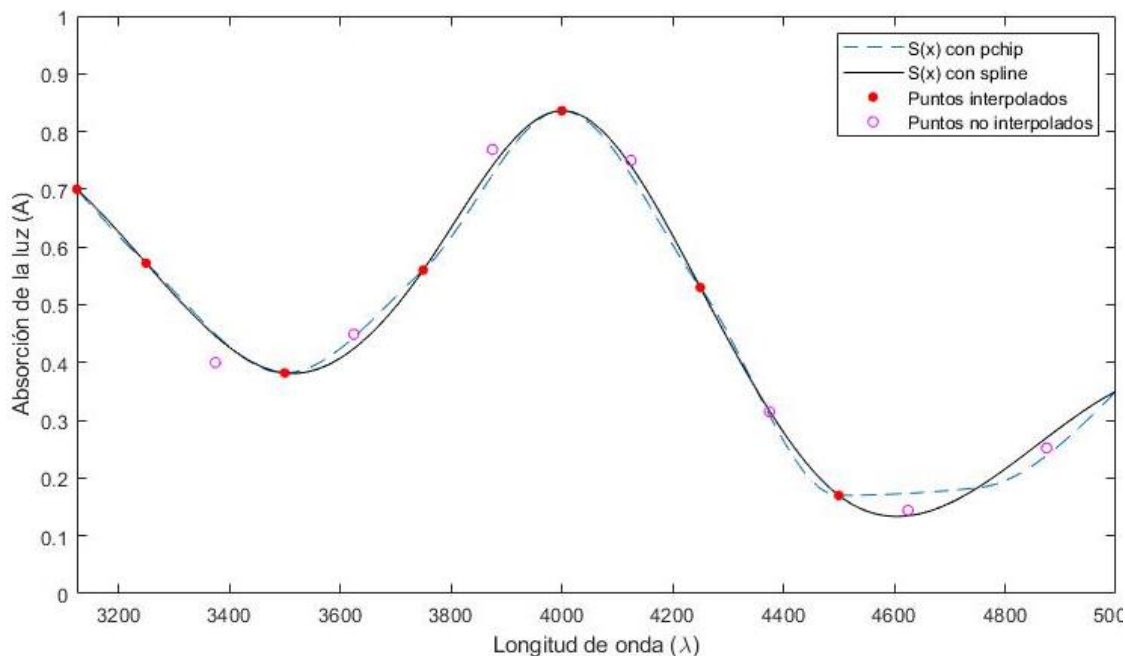


Fig. 3. Interpolaciones de nueve valores de la longitud de onda en función de la absorción de la luz de un dimero de d-tartrato de Vanadio usando `spline` y `pchip`.

Ahora se intercambiará la variable independiente λ por x para los siguientes casos:

a) $x = \lambda$

Para este caso no hay cambio en la variable. La interpolación es la misma que la presentada anteriormente en la Fig. 3.

b) $x = \lambda/1000$

Este caso equivale a hacer una dilatación de los datos (decreciendo en un factor de 1000), llevando a los datos independientes al intervalo $[3.125, 5]$ (aproximadamente $[3, 5]$). La interpolación cambia en su escala, pero conserva su forma, como se ve en la Fig. 4a.

c) $x = (\lambda - 4000)/1000$

Este caso combina el inciso anterior al hacer una dilatación, pero añadiendo previamente una traslación en el eje (en 4000 unidades respectivamente), llevando a los datos independientes al intervalo $[-0.875, 1]$ (aproximadamente $[-1, 1]$). La interpolación cambia en su escala, pero también conserva su forma, como se ve en la Fig 4b.

En cada caso se calcularon los errores absolutos correspondientes entre los datos no interpolados y su valor interpolado correspondiente para los dos $S(x)$ obtenidos. Se encontró que **los errores son los mismos bajo cualquiera de estas transformaciones de la variable dependiente**, para toda la expresión decimal completa que ofrece el programa, por lo que concluimos que estos cambios y escalamientos **no afectan la exactitud de $S(x)$** . Esto se puede deber a que solo realizamos traslaciones y escalamientos, mismas que son transformaciones afines del plano.

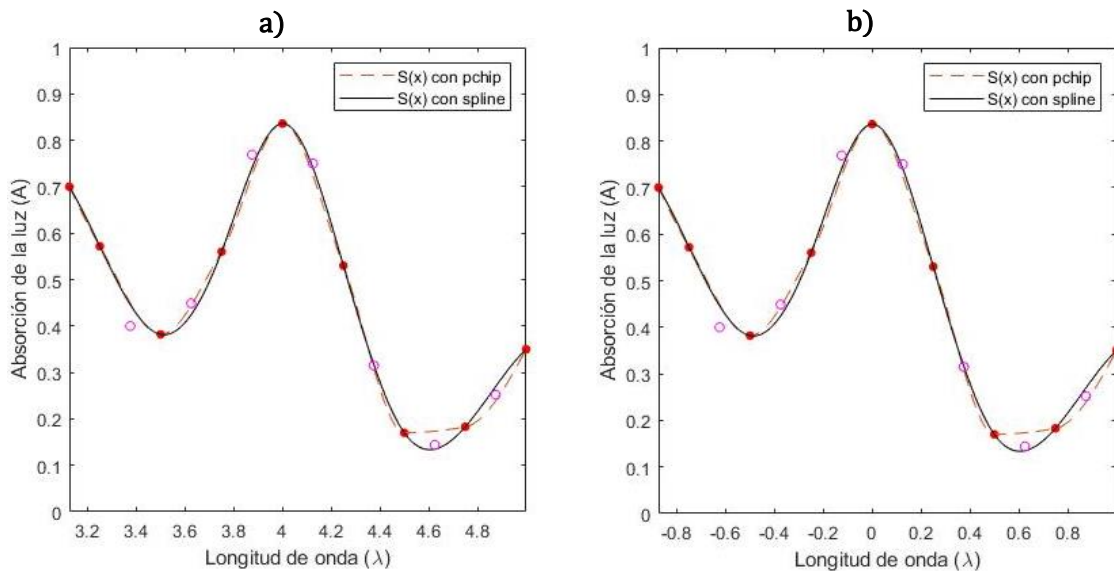


Fig. 4. Interpolaciones obtenidas para cambios de variable independiente. a) $x = \lambda/1000$.
b) $x = (\lambda - 4000)/1000$.

```
%Introducimos los datos de la longitud de onda y absorcion
l=[3125,3250:250:5000];
A=[0.7,0.572,0.382,0.56,0.836,0.53,0.17,0.183,0.35];
%Introducimos los datos restantes que no fueron interpolados
l_r=[3375:250:4875];
A_r=[0.4,0.449,0.769,0.75,0.315,0.144,0.252];
%ACTIVAR EL CAMBIO DE VARIABLE INDEPENDIENTE DESEADO a), b) o c)
% l=l; l_r=l_r;
% l=l./1000; l_r=l_r./1000;
% l=(l-4000)./1000; l_r=(l_r-4000)/1000;

%Interpolamos utilizando pchip y spline
pp_pchip=pchip(l,A);
pp_spline=spline(l,A);
%Graficamos la función interpolante en el intervalo correspondiente
x=l(1):(l(9)-l(1))/1000:l(9);
plot(x,ppval(pp_pchip,x),'--'); hold on;
plot(x,ppval(pp_spline,x),'k-'); hold on;
xlabel('Longitud de onda (\lambda)'); ylabel('Absorción de la luz (A)');
%Graficamos puntos interpolados y no interpolados
plot(l(1:1:7),A(1:1:7),'r.','markersize',15);
plot(l_r(1:1:7),A_r(1:1:7),'mo','markersize',5);
legend('S(x) con pchip','S(x) con spline',...
    'Puntos interpolados','Puntos no interpolados');
%Calculamos los errores de los no interpolados con su interpolación
err_spline=abs(A_r-ppval(pp_spline,l_r));
err_pchip=abs(A_r-ppval(pp_pchip,l_r));
disp('Cambio de variable independiente x=(lambda-4000)/1000');
disp('Errores absolutos entre puntos no interpolados y S(x) spline:');
disp(err_spline);
disp('Errores absolutos entre puntos no interpolados y S(x) pchip:');
disp(err_pchip);
```

```
Command Window

>> ejercicio_3_23
Cambio de variable independiente x=lambda
Errores absolutos entre puntos no interpolados y S(x) spline:
    0.046187    0.024656    0.028313    0.0095929    0.0011844    0.0092698    0.01777

Errores absolutos entre puntos no interpolados y S(x) pchip:
    0.04928    0.0050529    0.043947    0.025649    0.0063514    0.029485    0.012985

>> ejercicio_3_23
Cambio de variable independiente x=lambda/1000
Errores absolutos entre puntos no interpolados y S(x) spline:
    0.046187    0.024656    0.028313    0.0095929    0.0011844    0.0092698    0.01777

Errores absolutos entre puntos no interpolados y S(x) pchip:
    0.04928    0.0050529    0.043947    0.025649    0.0063514    0.029485    0.012985

>> ejercicio_3_23
Cambio de variable independiente x=(lambda-4000)/1000
Errores absolutos entre puntos no interpolados y S(x) spline:
    0.046187    0.024656    0.028313    0.0095929    0.0011844    0.0092698    0.01777

Errores absolutos entre puntos no interpolados y S(x) pchip:
    0.04928    0.0050529    0.043947    0.025649    0.0063514    0.029485    0.012985
```

3.25 The absorption of sound (at 20°C, 40% humidity) as a function of frequency, f , is

f	>20	>40	63	>100	200	>400	800
$A(f)$	0.008	0.030	0.070	0.151	0.359	0.592	0.935
f	>1250	2000	>4000	10000	>16000	>40000	>80000
$A(f)$	1.477	2.870	9.618	53.478	122.278	429.310	850.536

Use the cubic spline $S(X)$ to interpolate the nine indicated ($>$) points in the following two ways.

- The data as is.
- $\log f$ versus $\log A(f)$.

Which seems to be better?

Evidentemente los datos tienen separaciones muy grandes entre algunos de ellos, la aplicación del logaritmo los escala tanto para visualizarlos mejor como para trabajarlos con un menor error numérico. Se interpolaron los nueve datos señalados en ambas escalas y se graficaron junto con todos los puntos (es decir, junto con los no interpolados) para apreciar el comportamiento, como se muestra en la Fig. 5.

Evidentemente **la aplicación de la escala logarítmica parece ser mejor** pues muestra una distribución uniforme entre dato y dato, mientras que al no aplicarla hace que se junten muchos nodos al principio del intervalo. Para compararlas se calculó el error absoluto acumulado de los puntos no interpolados en cada caso:

- Usando spline, el error acumulado sin la escala es de 1.3271, aplicando la escala se obtiene 0.10873.
- Usando pchip, el error acumulado sin la escala es de 2.7453, aplicando la escala se obtiene 0.21039.

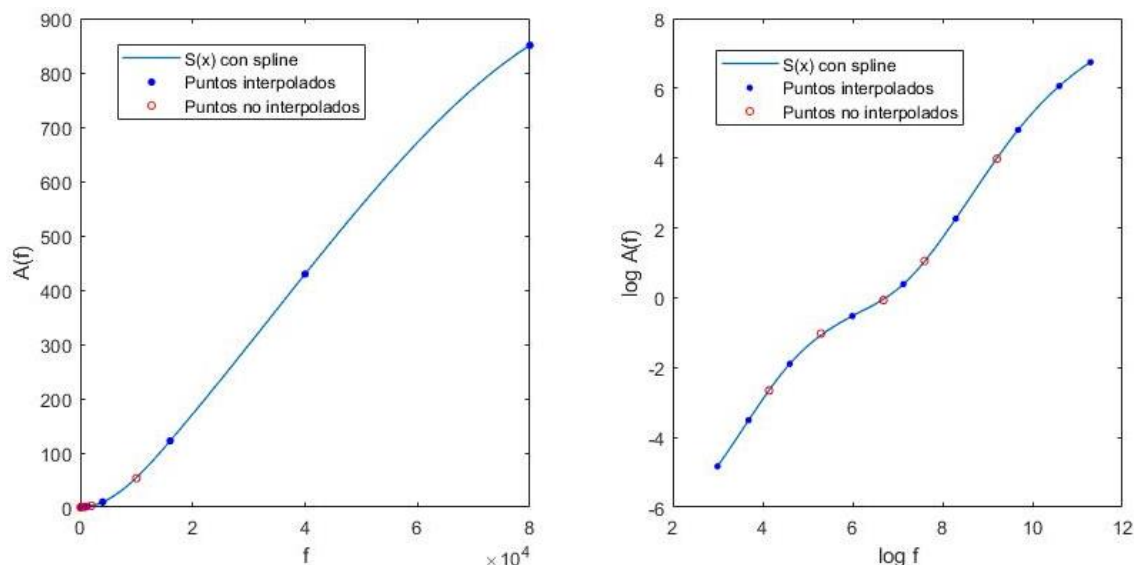


Fig. 5. Interpolación de datos de frecuencia sonora en función de su absorción. La aplicación de la escala logarítmica para datos muy separados permite interpolar con mayor exactitud.

3.27 In performing potentiometric titrations one obtains a potential difference curve plotted against volume of titrant added. The following table gives the measurements for the potentiometric titration of Fe^{2+} solution with 0.1095N Ce^{4+} solution using platinum and calomel electrodes.

Added sol. (ml)	1.0	5.0	10.0	15.0	20.0	21.0	22.0
E (mV)	373	415	438	459	491	503	523
Added sol. (ml)	22.5	22.6	22.7	22.8	22.9	23.0	23.1
E (mV)	543	550	557	565	575	590	620
Added sol. (ml)	23.2	23.3	23.4	23.5	24.0	26.0	30.0
E (mV)	860	915	944	958	986	1067	1125

Compute the cubic spline $S(x)$ for these data (use about 15 interpolating points). Graph $S(x)$ or x in $[20, 24]$. How well does it behave? The physical problem has exactly one inflection point. Is this true for $S(x)$?

Se pide elegir 15 puntos distintos para interpolar, por lo que se diseñó un programa para elegir aleatoriamente 15 de los 21 puntos disponibles sin repetir; esto se hizo con la función `randperm` (cuyo argumento es un entero k y devuelve un vector con una permutación aleatoria de los enteros del 1 al k). Finalmente se interpolaron los 15 datos con `pchip` y `spline` y se graficaron en el intervalo $[20, 24]$.

Se muestran las distintas gráficas obtenidas para cuatro ejecuciones distintas del programa en la Fig. 6. Se invita a ejecutar más veces el programa para corroborar los resultados presentados.

En general podemos concluir con las siguientes observaciones basadas en muchos experimentos numéricos y en la implementación de cada método de interpolación:

- 1) En la gran mayoría de experimentos, **la interpolación con splines cúbicos (`spline`) tiene comportamientos extraños**. Esto se debe a varios factores.
 - a. En primer lugar, la elección de los 15 puntos puede no corresponder a los del intervalo $[20, 24]$, por lo que el comportamiento de $S(x)$ en este intervalo no es exactamente el de los datos, más aún, los coeficientes del algoritmo de `spline` están estrechamente relacionados con todos los puntos, **por lo que la preservación de la forma de los datos se ve sumamente afectada al no tomar suficientes puntos**.
 - b. En segundo lugar, recordemos que una de las propiedades principales para el spline cúbico es que tiene menor curvatura local que cualquier otra función interpolante para los mismos puntos, **lo que da lugar a sus características oscilaciones entre nodo y nodo**. Recordemos que **`spline` funciona mejor para funciones suaves** (cosa que no verifican los datos experimentales).
 - c. Como consecuencia de los hechos anteriores, podemos explicar por qué **$S(x)$ tiene más de un punto de inflexión en la mayoría de los experimentos**.

2) En la gran mayoría de experimentos, la interpolación con *Interpolación de Hermite Polinómica Cúbica a Trozos (pchip)* tiene un comportamiento común y muy parecido a los datos experimentales. Esto se explica considerando que

- A pesar de que la elección de los 15 puntos es aleatoria, una de las propiedades de pchip es que, al calcular el polinomio cúbico adecuado en cada subintervalo, toma como información la derivada, aproximada como la pendiente entre los extremos del intervalo, es decir, un par de puntos experimentales. **Esto hace que preserve la forma, es decir, la monotonidad de los datos.**
- A diferencia de spline, pchip no tiene oscilaciones y además, por las condiciones antes mencionadas, asegura que la primera derivada es continua. Esto implica que, si los datos tienen un extremo local, también se verá reflejado por este tipo de interpolación. **Por este motivo (junto con los anteriores) $S(x)$ tiene un solo punto de inflexión en cada caso.**

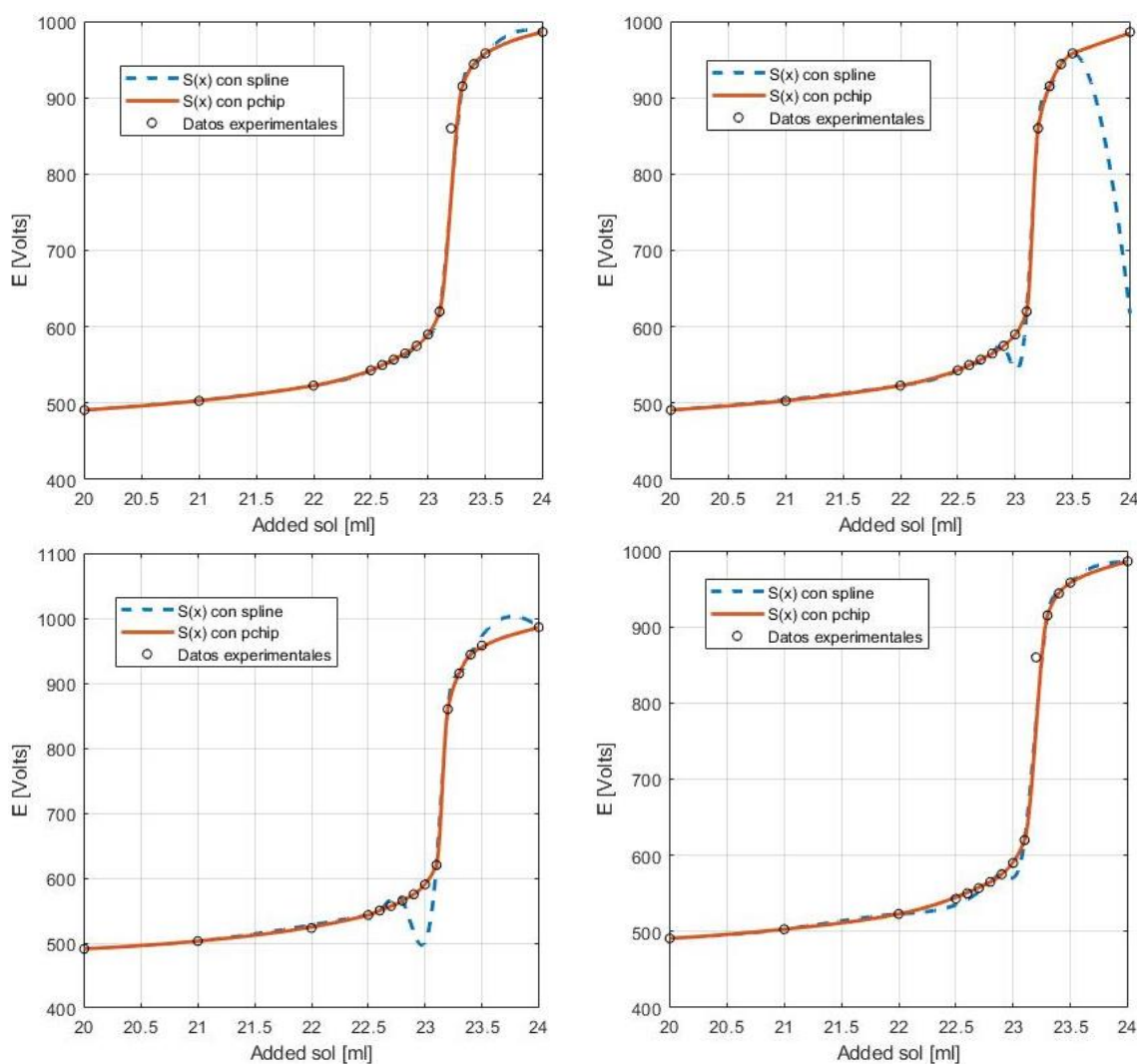


Fig. 6. Interpolaciones de una muestra de 15 datos aleatorios del problema obtenidas con las funciones spline y pchip. Se muestran cuatro experimentos arbitrarios.

```
%Introducimos los 21 datos reales
s_total=[1,5,10,15,20,21,22,22.5,22.6,22.7,22.8,...
        22.9,23,23.1,23.2,23.3,23.4,23.5,24,26,30];
E_total=[373,415,438,459,491,503,523,543,550,557,...
        565,575,590,620,860,915,944,958,986,1067,1125];
%Seleccionamos aleatoriamente 15 de ellos para interpolar
indices=randperm(21);
s=s_total(indices(1:15));
E=E_total(indices(1:15));
%Interpolamos usando spline y pchip
pp_spline=spline(s,E); pp_pchip=pchip(s,E);
%Graficamos en el intervalo [20,24]
x=20:0.01:24;
plot(x,ppval(pp_spline,x),'--','linewidth',2); hold on;
plot(x,ppval(pp_pchip,x),'-','linewidth',2); hold on;
plot(s_total(5:19),E_total(5:19),'ko','markersize',5);
%Propiedades de gráfica
xlabel('Added sol [ml]'); ylabel('E [Volts]'); axis square; grid on;
legend('S(x) con spline','S(x) con pchip', 'Datos experimentales');
```

3.28 The potential energy of two or more interacting molecules is called van der Waal's interaction energy. A theoretical calculation for two interacting helium atoms has the set of energies $V(r)$ for various values of the internuclear distance r given below. The energy exhibits repulsion ($V > 0$) for small r and attraction ($V < 0$) for larger values of r .

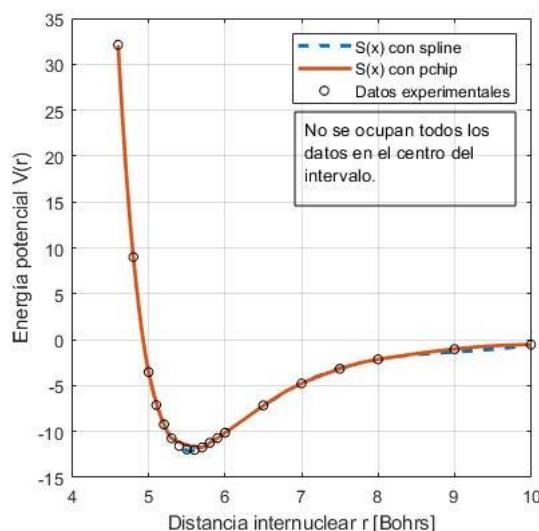
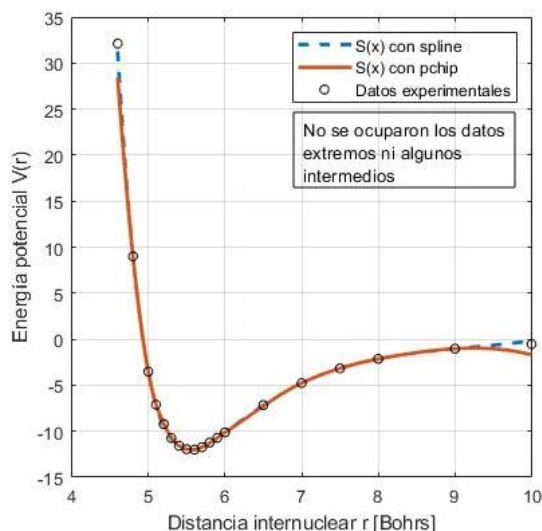
r [Bohrs]	4.6	4.8	5.0	5.1	5.2
$V(r)$	32.11	9.00	-3.52	-7.11	-9.22
r [Bohrs]	5.3	5.4	5.5	5.6	5.7
$V(r)$	-10.74	-11.57	-11.95	-12.00	-11.73
r [Bohrs]	5.8	5.9	6.0	6.5	7.0
$V(r)$	-11.23	-10.71	-10.13	-7.15	-4.77
r [Bohrs]	7.5	8.0	9.0	10.0	
$V(r)$	-3.17	-2.14	-1.03	-0.54	

Compute the cubic spline $S(x)$ using about 12 interpolating points. How well does it work?

La selección aleatoria de los 12 puntos a interpolar implica que para distintos experimentos, las interpolaciones concuerden de buena o mala manera con todos los datos (interpolados o no) en general.

Al igual que en el ejercicio anterior, se hace relevante el comportamiento propio de las interpolaciones obtenidas con `pchip` y `spline`. Mientras que la `pchip` intenta preservar de la mejor manera la forma de los datos (consiguiéndolo en la mayoría de experimentos, pero como obviamente depende de ellos suele fallar), `spline` es muy susceptible a tener oscilaciones entre punto y punto; sin embargo, en muchos experimentos consigue mantener la forma de los datos incluso mejor de `pchip`. Este último hecho puede deberse a la implementación de los métodos, en particular la relación que guardan las condiciones de frontera (en algunos experimentos es evidente como los ajustes se separan tras interpolar el último punto).

Como comentario final, en la mayoría de los experimentos (Fig. 7) ambos métodos consiguen interpolar de manera aceptable a todos los puntos. Para el caso de tomar todos los puntos las interpolaciones son esencialmente idénticas.



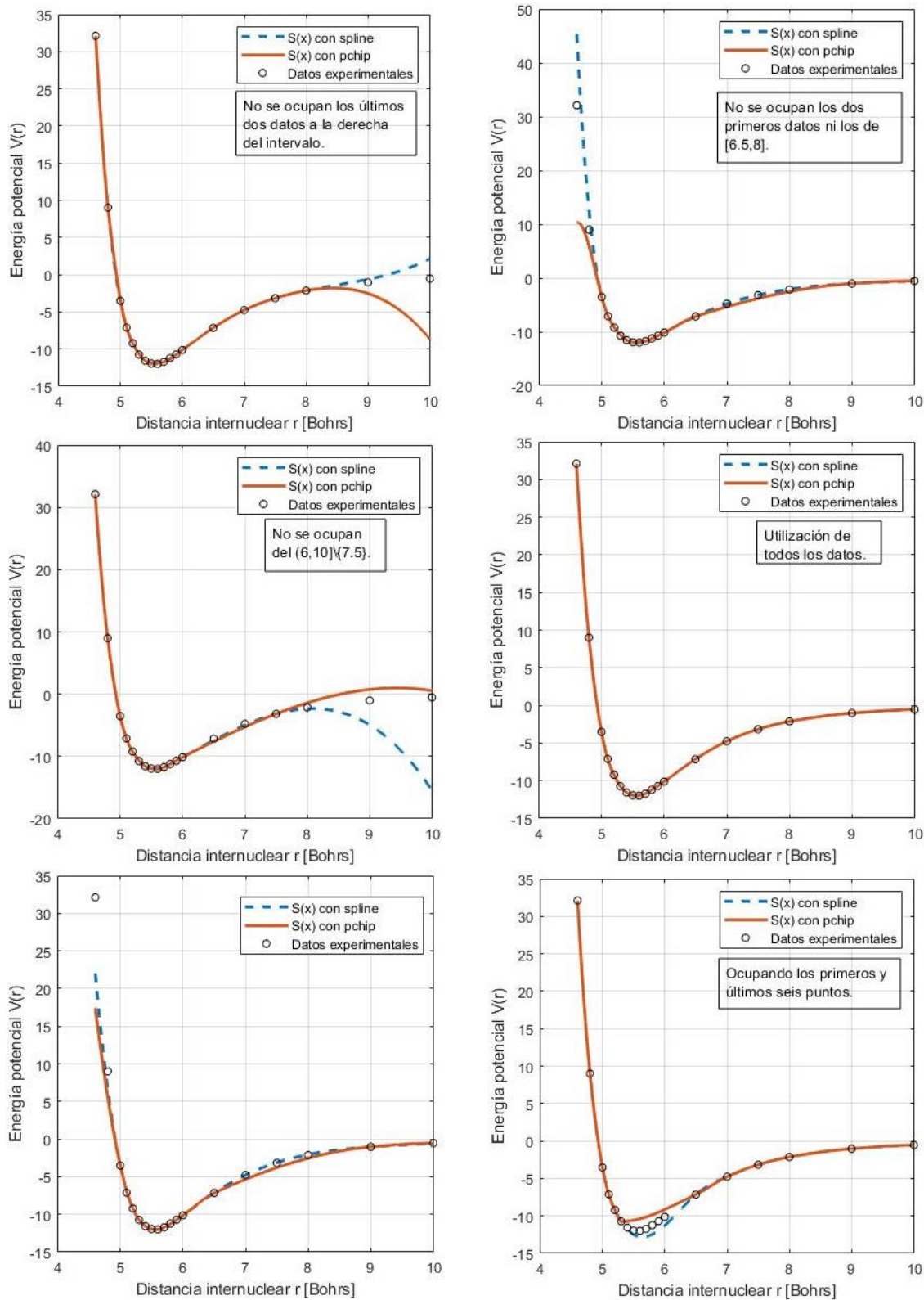


Figura 7. Interpolaciones de muestras aleatorias de 12 datos de la interacción de Van der Waals entre dos átomos de Helio.

```
%Introducimos los 19 datos reales
s_total=[4.6,4.8,5:0.1:6,6.5,7,7.5,8,9,10];
E_total=[32.11,9,-3.52,-7.11,-9.22,-10.74,-11.57,-11.95,-12,...
        -11.73,-11.23,-10.71,-10.13,-7.15,-4.77,-3.17,-2.14,-1.03,-0.54];
%Seleccionamos aleatoriamente 12 de ellos para interpolar
indices=randperm(19);
s=s_total(indices(1:12));
E=E_total(indices(1:12));
%Interpolamos usando spline y pchip
pp_spline=spline(s,E); pp_pchip=pchip(s,E);
%Graficamos en el intervalo [4.6,10]
x=4.6:0.005:10;
plot(x,ppval(pp_spline,x),'--','linewidth',2); hold on;
plot(x,ppval(pp_pchip,x),'-','linewidth',2); hold on;
plot(s_total(1:19),E_total(1:19),'ko','markersize',5);
%Propiedades de gráfica
xlabel('Distancia internuclear r [Bohrs]');
ylabel('Energía potencial V(r)'); axis square; grid on;
legend('S(x) con spline','S(x) con pchip', 'Datos experimentales');
```

3.30 In an extra reference, a method is given for deducing the diffusion coefficient D for chloroform in polystyrene from uptake measurements. Using several assumptions, they arrive at the quantity

$$\widehat{D}(C_0) = \frac{1}{C_0} \int_0^{C_0} D(C) dC$$

which can be measured for a number of C_0 values. A differentiation with respect to C_0 gives an expression for D in terms of the quantity $D = \frac{d}{dC_0} [C_0 \widehat{D}(C_0)]$. Using the data

C_0	5.0	7.5	9.9	12.9
$\widehat{D}(C_0)$	0.0240	0.0437	0.0797	0.1710
C_0	13.2	15.1	16.3	16.8
$\widehat{D}(C_0)$	0.1990	0.3260	0.8460	0.9720

approximate D for each C_0 value by differentiating the appropriate spline fit.

De la expresión $D = \frac{d}{dC_0} [C_0 \widehat{D}(C_0)]$ tenemos que $D = \frac{d}{dC_0} [C_0 \widehat{D}(C_0)] = \widehat{D}(C_0) + C_0 \widehat{D}'(C_0)$. Para aproximar los valores de D en cada punto C_0 se interpolaron los datos dados, obteniendo la función $S(x)$ que es una aproximación de $\widehat{D}(C_0)$, por lo que la denotaremos de esta manera.

Las funciones spline y pchip proporcionan polinomios a trozos de orden cúbico y además **aseguran la continuidad de la derivada de la función interpolante en todo el intervalo**. Se grafican sus respectivas interpolaciones en la Fig. 8.

Recordemos que para n puntos $\{(x_i, y_i)\}_{i=1}^n$, estas funciones interpolan a trozos de la forma:

$$S(x) = \begin{cases} a_1(x - x_1)^3 + b_1(x - x_1)^2 + c_1(x - x_1) + d_1 & \text{si } x_1 \leq x \leq x_2 \\ a_2(x - x_2)^3 + b_2(x - x_2)^2 + c_2(x - x_2) + d_2 & \text{si } x_2 \leq x \leq x_3 \\ \vdots & \vdots \\ a_{n-1}(x - x_{n-1})^3 + b_{n-1}(x - x_{n-1})^2 + c_{n-1}(x - x_{n-1}) + d_{n-1} & \text{si } x_{n-1} \leq x \leq x_n \end{cases}$$

Por lo que se verifica $S'(x) = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i$ para $x_i \leq x \leq x_{i+1}$ para valores de $i = 1, \dots, n - 1$.

Recordemos que $S(x) \approx \widehat{D}(C_0)$, por lo tanto $D(C_0) \approx \widehat{D}(C_0) + C_0 \widehat{D}'(C_0) \approx S(C_0) + C_0 S'(C_0)$, pero los puntos que consideramos en la interpolación fueron $\{(C_{0i}, \widehat{D}(C_{0i}))\}_{i=1}^8$, por lo que

$$\begin{aligned} D(C_{0i}) &\approx a_i(C_{0i} - C_{0i})^3 + b_i(C_{0i} - C_{0i})^2 + c_i(C_{0i} - C_{0i}) + d_i \\ &\quad + C_0(3a_i(C_{0i} - C_{0i})^2 + 2b_i(C_{0i} - C_{0i}) + c_i) = 0 + 0 + 0 + d_i + C_0(0 + 0 + c_i) \\ &\therefore D(C_{0i}) \approx d_i + c_i C_0 \quad \text{para } i = 1, \dots, 7 \end{aligned}$$

Para el último punto es un caso especial, aquí no se anulan los términos por lo que en este caso

$$\therefore D(C_{08}) = a_7(h^3 + 3C_{08}h^2) + b_7(h^2 + 2C_{08}h) + c_7(h + C_{08}) + d_7 \quad \text{con } h = C_{08} - C_{07}$$

La estructura que producen las funciones `spline` y `pchip` contienen todos los coeficientes requisitados $\{a_i, b_i, c_i, d_i\}_{i=1}^7$, por lo que en el programa **simplemente se calculan los D correspondientes a cada C_0 con las expresiones que se determinaron anteriormente.** Denominaremos este procedimiento como *Método propio*.

Para verificar los resultados anteriores se utilizó la función `fnder` de Matlab. Esta función recibe la estructura de polinomio a trozos devuelta por `spline` o `pchip` y devuelve otra estructura del mismo estilo, pero con la información de los coeficientes de la derivada, lista para usarse con la función `ppval` que utilizamos para evaluar dichas estructuras. Esta función es especial para encontrar derivadas de interpolaciones en Matlab, incluso si no son polinómicas. Entonces **se utilizó la expresión $D(C_0) \approx \hat{D}(C_0) + C_0 \hat{D}'(C_0) \approx S(C_0) + C_0 S'(C_0)$ directamente, obteniendo los respectivos D .** Denominaremos este procedimiento como *Método fnder*.

Se muestran los valores correspondientes a cada método y se comprueba que coinciden en todos los valores, concluyendo que los valores encontrados para D son correctos bajo cualquier método y bajo cualquier método de interpolación.

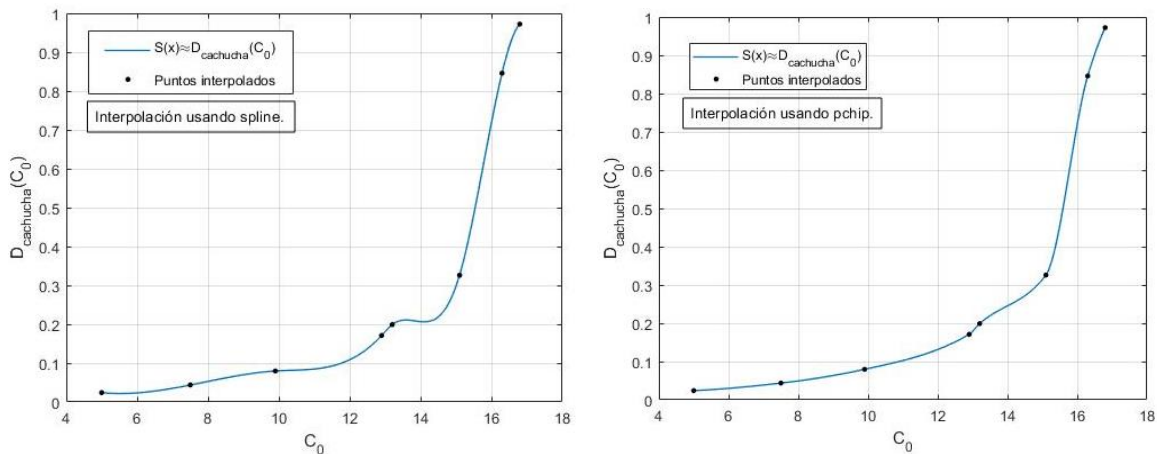


Fig. 8. Interpolaciones de los puntos $(C_0, \hat{D}(C_0))$ utilizando `spline` y `pchip`.

```

Command Window
>> clear
>> ejercicio_3_30
Interpolación usando SPLINE
(Método propio). Valores calculados de D para cada c_0:
    -0.02077    0.17854    0.13726    1.447    1.1354    4.9613    7.0424    2.5291

(Método fnder). Valores calculados de D para cada c_0:
    -0.02077    0.17854    0.13726    1.447    1.1354    4.9613    7.0424    2.5291

>> ejercicio_3_30
Interpolación usando PCHIP
(Método propio). Valores calculados de D para cada c_0:
    0.045237    0.12136    0.27617    0.85838    1.2702    2.1769    5.8584    4.3096

(Método fnder). Valores calculados de D para cada c_0:
    0.045237    0.12136    0.27617    0.85838    1.2702    2.1769    5.8584    4.3096

```



```
%Introducimos los 8 datos dados
c_0=[5,7.5,9.9,12.9,13.2,15.1,16.3,16.8];
D_cachucha=[0.024,0.0437,0.0797,0.1710,0.1990,0.326,0.846,0.972];
%Interpolamos y recuperamos las estructuras que contienen los coefs
%SELECCIONAR TIPO DE INTERPOLACIÓN
pp=spline(c_0,D_cachucha);
% pp=pchip(c_0,D_cachucha);

%METODO PROPIO, HALLAR LAS DERIVADAS MANUALMENTE
%Calculamos los 7 primeros valores de D
D=zeros(1,8);
for i=1:7
    %Recuperamos los coeficientes a_i, b_i c_i y d_i
    a=pp.coefs(i,1); b=pp.coefs(i,2); c=pp.coefs(i,3); d=pp.coefs(i,4);
    %Computamos el valor de D
    D(i)=d+c*c_0(i);
end
%Computamos el caso especial en el ultimo punto
h=c_0(8)-c_0(7);
D(8)=a*(h^3+3*c_0(8)*h^2)+b*(h^2+2*c_0(8)*h)+c*(h+c_0(8))+d;
%Mostramos los resultados de los D para cada c_0
disp('(Método propio). Valores calculados de D para cada c_0:'); disp(D);

%METODO USANDO LA FUNCION fnder DE MATLAB
%Esta función recibe la estructura pp de la interpolación
%Devuelve otra estructura pp correspondiente a la derivada
D=ppval(pp,c_0)+c_0.*ppval(fnder(pp),c_0);
%Mostramos los resultados de los D para cada c_0
disp('(Método fnder). Valores calculados de D para cada c_0:'); disp(D);

%Gráfica auxiliar
x=5:0.01:16.8;
plot(x,ppval(pp,x),'linewidth',1); hold on;
xlabel('C_0'); ylabel('D_{cachucha}(C_0)'); grid on;
plot(c_0(1:8),D_cachucha(1:8),'k.','markersize',10);
legend('S(x)\approx D_{cachucha}(C_0)','Puntos interpolados');
```


3.31 Show that the cubic spline $S(x)$ has a critical point z in $[x_i, x_{i+1}]$, that is, $S'(z) = 0$, if and only if the following are true:

- I. $z = x_i + \left(-c_i \pm \sqrt{c_i^2 - 3b_id_i} \right) / (3d_i)$
- II. $x_i \leq z$
- III. $z \leq x_{i+1}$

Why is it not sufficient merely to use (I) and the test $b_ib_{i+1} = S'(x_i)S'(x_i + 1) < 0$?

Escribamos la expresión de un spline cúbico arbitrario para n puntos como

$$S(x) = \begin{cases} d_1(x - x_1)^3 + c_1(x - x_1)^2 + b_1(x - x_1) + a_1 & \text{si } x_1 \leq x \leq x_2 \\ d_2(x - x_2)^3 + c_2(x - x_2)^2 + b_2(x - x_2) + a_2 & \text{si } x_2 \leq x \leq x_3 \\ \vdots & \vdots \\ d_{n-1}(x - x_{n-1})^3 + c_{n-1}(x - x_{n-1})^2 + b_{n-1}(x - x_{n-1}) + a_{n-1} & \text{si } x_{n-1} \leq x \leq x_n \end{cases}$$

Una de las condiciones de los splines es que la primera derivada sea continua, por lo que podemos expresar sin pérdida de generalidad que

$$S'(x) = \begin{cases} 3d_1(x - x_1)^2 + 2c_1(x - x_1) + b_1 & \text{si } x_1 \leq x \leq x_2 \\ 3d_2(x - x_2)^2 + 2c_2(x - x_2) + b_2 & \text{si } x_2 \leq x \leq x_3 \\ \vdots & \vdots \\ 3d_{n-1}(x - x_{n-1})^2 + 2c_{n-1}(x - x_{n-1}) + b_{n-1} & \text{si } x_{n-1} \leq x \leq x_n \end{cases}$$

Dicho esto, supongamos que $z \in [x_i, x_{i+1}]$ es un punto crítico de $S(x)$, es decir $S'(z) = 0$. Trivialmente se verifican las condiciones (I) y (II) de forma unívoca.

Veamos que $S'(z) = 0$ implica que $S'(z) = 3d_i(z - x_i)^2 + 2c_i(z - x_i) + b_i = 0$, desarrollando:

$$\begin{aligned} 3d_i(z - x_i)^2 + 2c_i(z - x_i) + b_i &= 0 \\ \Leftrightarrow 3d_i(z^2 - 2zx_i + x_i^2) + 2c_i(z - x_i) + b_i &= 0 \\ \Leftrightarrow 3d_iz^2 - 6d_ix_iz + 3d_ix_i^2 + 2c_iz - 2c_ix_i + b_i &= 0 \\ \Leftrightarrow z^2(3d_i) + z(2c_i - 6d_ix_i) + (3d_ix_i^2 - 2c_ix_i + b_i) &= 0 \end{aligned}$$

Que es una ecuación cuadrática para z , esta ecuación tiene solución si y sólo si

$$\begin{aligned} z &= \frac{6d_ix_i - 2c_i \pm \sqrt{(2c_i - 6d_ix_i)^2 - 4(3d_i)(3d_ix_i^2 - 2c_ix_i + b_i)}}{6d_i} \\ \Leftrightarrow z &= \frac{6d_ix_i - 2c_i \pm \sqrt{4c_i^2 - 24c_id_ix_i + 36d_i^2x_i^2 - 36d_i^2x_i^2 + 24d_ic_ix_i - 12d_ib_i}}{6d_i} \\ \Leftrightarrow z &= \frac{6d_ix_i - 2c_i \pm \sqrt{4c_i^2 - 12d_ib_i}}{6d_i} = \frac{3d_ix_i - c_i \pm \sqrt{c_i^2 - 3d_ib_i}}{3d_i} = x_i - \frac{c_i \pm \sqrt{c_i^2 - 3d_ib_i}}{3d_i} \\ \therefore z &= x_i - \left(c_i \pm \sqrt{c_i^2 - 3d_ib_i} \right) / (3d_i) \end{aligned}$$

Por lo tanto, que $z \in [x_i, x_{i+1}]$ es un punto crítico de $S(x)$ si y sólo si se verifican (I), (II) y (III) ■

La condición $b_i b_{i+1} = S'(x_i)S'(x_{i+1}) < 0$ nos dice que **$S'(x)$ cambia de signo en los extremos del intervalo** que estamos considerando, $[x_i, x_{i+1}]$, sin embargo, consideremos el siguiente par de casos:

- Supongamos que **$S'(x_i) = 0$ y $S'(x_{i+1}) = 0$** .
En esta situación $S'(x_i)S'(x_{i+1}) = 0$ y tenemos que x_i y x_{i+1} son puntos críticos de $S(x)$, sin embargo, al fallar la condición $S'(x_i)S'(x_{i+1}) < 0$ concluiríamos que no hay puntos críticos en $[x_i, x_{i+1}]$.
- Supongamos que **$S'(x)$ cambia dos veces de signo en (x_i, x_{i+1})** .
En esta situación necesariamente $S'(x_i)S'(x_{i+1}) > 0$ pues al haber dos intersecciones con el eje x , los signos de los extremos del intervalo son los mismos, positivos o negativos, esto hace fallar la condición $S'(x_i)S'(x_{i+1}) < 0$ ya que concluiríamos que no hay puntos críticos en $[x_i, x_{i+1}]$.

Por esto no podemos meramente aplicar la prueba $S'(x_i)S'(x_{i+1}) < 0$ y (I), ya que **no detecta todos los casos** para los puntos críticos.

3.32 Show that the cubic spline $S(x)$ has an inflection point z in (x_i, x_{i+1}) , that is, $S''(x) = 0$, if and only if $c_i c_{i+1} < 0$, in which case $z = x_i - c_i/3d_i$.

Utilizando el problema anterior, específicamente las expresiones para $S(x)$ y $S'(x)$, escribamos:

$$S''(x) = \begin{cases} 6d_1(x - x_1) + 2c_1 & \text{si } x_1 \leq x \leq x_2 \\ 6d_2(x - x_2) + 2c_2 & \text{si } x_2 \leq x \leq x_3 \\ \vdots & \vdots \\ 6d_{n-1}(x - x_{n-1}) + 2c_{n-1} & \text{si } x_{n-1} \leq x \leq x_n \end{cases}$$

Consideremos $z \in (x_i, x_{i+1})$. z es un punto de inflexión si y sólo si $S''(z) = 0$, esto se verifica si y sólo si $z = x_i - c_i/3d_i$.

Un **punto de inflexión** es un punto en el que **la curvatura de la función cambia de signo**. Recordemos que para una función $f(x)$, su curvatura local en x viene dada por

$$\kappa(x) = \frac{f''(x)}{\left(1 + (f'(x))^2\right)^{\frac{3}{2}}}$$

es por esto por lo que buscamos los puntos donde $f''(x) = 0$ pues señalan la transición de la función a ser cóncava o convexa. Notemos que **la curvatura de $S(x)$ solo cambia de signo si $S''(x)$ cambia de signo**, ya que el denominador de la expresión siempre es positivo.

Ciertamente, nuestra expresión de $S''(x)$ es una función a trozos donde en cada intervalo tenemos un término lineal. Esto implica que **a lo más hay un punto en cada intervalo $[x_i, x_{i+1}]$** , en particular (x_i, x_{i+1}) , **donde $S''(x)$ cambia de signo**, o lo que es lo mismo, $S''(x) = 0$. Ya hemos demostrado que este punto es $z = x_i - c_i/3d_i$ con $z \in (x_i, x_{i+1})$.

Entonces **un criterio para detectar este cambio de signo en $S''(x)$ es analizar lo que pasa en los extremos del intervalo (x_i, x_{i+1})** (no es necesario preocuparse por más de un punto por lo expuesto en el argumento anterior). Si hay tal cambio entonces se debe cumplir que

$$\begin{aligned} S''(x_i)S''(x_{i+1}) &< 0 \\ \Leftrightarrow (6d_i(x_i - x_i) + 2c_i)(6d_{i+1}(x_{i+1} - x_{i+1}) + 2c_{i+1}) &< 0 \\ \Leftrightarrow (0 + 2c_i)(0 + 2c_{i+1}) &< 0 \\ \Leftrightarrow (2c_i)(2c_{i+1}) &< 0 \\ \Leftrightarrow 4c_i c_{i+1} &< 0 \\ \Leftrightarrow c_i c_{i+1} &< 0 \end{aligned}$$

Por lo tanto, concluimos que **$S(x)$ tiene un punto de inflexión en $z = x_i - c_i/3d_i$ con $z \in (x_i, x_{i+1})$ si y sólo si $S''(z) = 0$ o lo que es lo mismo, $S''(x)$ cambia de signo en (x_i, x_{i+1}) , hecho que se da si y sólo si $c_i c_{i+1} < 0$ ■**