3.1 ¿El polinomio interpolante de Lagrange siempre tiene grado exacto de $n-1$? Si no, ilustre con un ejemplo.

No, puede tener un grado menor.

Consideremos el problema de interpolar la función $f(x) = \sin x$ en los puntos $\{(0,0), (\pi, 0), (2\pi, 0)\}$. El polinomio interpolante de Lagrange es $p(x) = y_1 L_1 + y_2 L_2 + y_3 L_3$. Obteniendo los L_1, L_2 y L_3:

$$L_1 = \left(\frac{x-\pi}{0-\pi}\right)\left(\frac{x-2\pi}{0-2\pi}\right) = \left(\frac{\pi - x}{\pi}\right)\left(\frac{2\pi - x}{\pi}\right)$$

$$L_2 = \left(\frac{x-0}{\pi-0}\right)\left(\frac{x-2\pi}{\pi-2\pi}\right) = \left(\frac{x}{\pi}\right)\left(\frac{2\pi - x}{\pi}\right)$$

$$L_2 = \left(\frac{x-0}{2\pi-0}\right)\left(\frac{x-\pi}{2\pi-\pi}\right) = \left(\frac{x}{2\pi}\right)\left(\frac{x-\pi}{\pi}\right)$$

Sin embargo, obtenemos $p(x) = 0 \cdot L_1 + 0 \cdot L_2 + 0 \cdot L_3 = 0$ ya que $y_i = 0, i = 1,2,3$. Por lo tanto, el polinomio interpolante obtenido es el polinomio cero, y dado que tenemos 3 puntos, obviamente tiene grado menor que 2. En la Figura 1 se grafica $\sin x$ y $p(x)$.

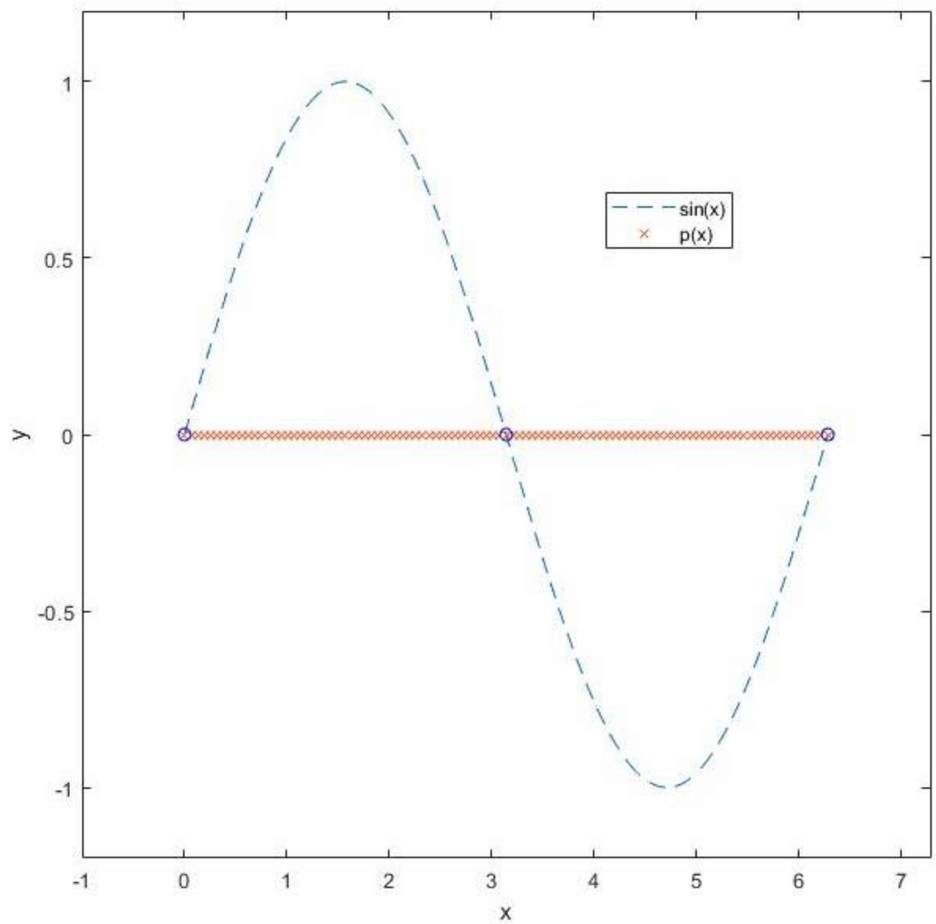

Figura 1. $\sin x$ y el polinomio interpolante de Lagrange, $p(x) = 0$.

3.2 Suppose that $f(x)$ is a polynomial of degree $n-1$ or less. Prove that if $P_n(x)$ interpolates $f(x)$ at n distinct points, then $P_n(x) \equiv f(x)$. Make up an example $(n \geq 3)$ and verify by direct calculation.

Dado que $f(x)$ es un polinomio de grado $n-1$, entonces tiene n derivadas continuas, ya que sus derivadas también son polinomios, más aún, $f^{(i)}(x)$ es un polinomio de grado $n-1-i$, para $i = 1, \cdots, n-1$. Notemos que $f^{(n)}(x) = 0$.

Se pide interpolar n puntos de $f(x)$, éstos tienen la forma $(x_i, y_i) = (x_i, f(x_i))$, $i = 1, \cdots, n$ y verifican que $x_i \neq x_j, \forall\ i \neq j$.

Utilizando el teorema del error del polinomio interpolante:

Supongamos que los valores de interpolación y_i, $i = 1, \cdots, n$ son los valores de interpolación exactos de una función $y_i = f(x_i)$, $i = 1, \cdots, n$ y si además la función tiene n derivadas continuas entonces

$$f(x) - p(x) = \frac{f^{(n)}(\xi)}{n!} \prod_{i=1}^{n} (x - x_i), \quad para\ algún\ \xi \in [x_1, x_n]$$

Veamos que en nuestro caso

$$f(x) - p(x) = \frac{f^{(n)}(\xi)}{n!} \prod_{i=1}^{n} (x - x_i) = \frac{0}{n!} \prod_{i=1}^{n} (x - x_i) = 0$$

Por lo tanto, concluimos que $f(x) \equiv p(x)$ ■

Como ejemplo, consideremos $f(x) = x^3 - 2x^2 + 2$ y los puntos $(-1,-1)$, $(0,2)$, $(1,1)$ y $(2,2)$. Calculando el polinomio interpolante de Lagrange tenemos que

$$L_1 = \left(\frac{x-0}{-1-0}\right)\left(\frac{x-1}{-1-1}\right)\left(\frac{x-2}{-1-2}\right) = \left(\frac{x}{-1}\right)\left(\frac{x-1}{-2}\right)\left(\frac{x-2}{-3}\right) = -\frac{1}{6}(x)(x-1)(\mathrm{x}-2)$$
$$L_2 = \left(\frac{x-(-1)}{0-(-1)}\right)\left(\frac{x-1}{0-1}\right)\left(\frac{x-2}{0-2}\right) = \left(\frac{x+1}{1}\right)\left(\frac{x-1}{-1}\right)\left(\frac{x-2}{-2}\right) = \frac{1}{2}(x+1)(x-1)(x-2)$$
$$L_3 = \left(\frac{x-(-1)}{1-(-1)}\right)\left(\frac{x-0}{1-0}\right)\left(\frac{x-2}{1-2}\right) = \left(\frac{x+1}{2}\right)\left(\frac{x}{1}\right)\left(\frac{x-2}{-1}\right) = -\frac{1}{2}(x+1)(x)(\mathrm{x}-2)$$
$$L_4 = \left(\frac{x-0}{2-0}\right)\left(\frac{x-1}{2-1}\right)\left(\frac{x-(-1)}{2-(-1)}\right) = \left(\frac{x}{2}\right)\left(\frac{x-1}{1}\right)\left(\frac{x+1}{3}\right) = \frac{1}{6}(x)(x-1)(\mathrm{x}+1)$$

$$\therefore p(x) = y_1L_1 + y_2L_2 + y_3L_3 + y_4L_4 = -L_1 + 2L_2 + L_3 + 2L_4$$
$$= \frac{1}{6}(\mathrm{x})(\mathrm{x}-1)(\mathrm{x}-2) + (\mathrm{x}+1)(\mathrm{x}-1)(\mathrm{x}-2) - \frac{1}{2}(\mathrm{x}+1)(\mathrm{x})(\mathrm{x}-2) + \frac{1}{3}(\mathrm{x})(\mathrm{x}-1)(\mathrm{x}+1)$$
$$= x^3 - 2x^2 + 2$$

Concluyendo que $f(x) = p(x)$, verificando en enunciado.

3.4 An alternate method for computing $P_n(x)$ is to write

$$p_n(x) = c_1 + c_2 x + c_3 x^2 + c_n x^{n-1}$$

Then the interpolation conditions, $p_n(x_i) = f_i$ for $1 \leq i \leq n$, yield a system of n equations in the n unknowns $c_1, \cdots, c_n$ that can be solved using the codes Factor/Solve. Unfortunately, there are two difficulties with this method: (1) it is expensive ($n^3/3$ multiplications), and (2) the coefficient matrix can be very ill-conditioned.

a) Implement this algorithm.
b) Test it on the data

x	5	10	15	20	30	40
$V(x)$	1.226	1.498	1.822	2.138	2.662	2.840

x	50	60	70	80	90	100
$V(x)$	2.807	2.542	2.210	1.877	1.539	1.201

Use the same six interpolating values and evaluate (see the next exercise) at the remaining points. What is COND? How do the answers compare with those in the text for&(x) computed by another method?

Los puntos para interpolar son $\left\{\begin{pmatrix}10\\1.498\end{pmatrix}, \begin{pmatrix}20\\2.138\end{pmatrix}, \begin{pmatrix}40\\2.840\end{pmatrix}, \begin{pmatrix}60\\2.542\end{pmatrix}, \begin{pmatrix}80\\1.877\end{pmatrix}, \begin{pmatrix}100\\1.201\end{pmatrix}\right\}$ denotados por $\left\{\begin{pmatrix}x_1\\y_1\end{pmatrix}, \cdots, \begin{pmatrix}x_6\\y_6\end{pmatrix}\right\}$ respectivamente. Para cumplir con la condición de interpolación $p(x_i) = y_i$ con $p(x) = c_1 + c_2 x + \cdots + c_6 x^5$ se obtiene el sistema lineal

$$\begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 & x_1^4 & x_1^5 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 & x_2^5 \\ 1 & x_3 & x_3^2 & x_3^3 & x_3^4 & x_3^5 \\ 1 & x_4 & x_4^2 & x_4^3 & x_4^4 & x_4^5 \\ 1 & x_5 & x_5^2 & x_5^3 & x_5^4 & x_5^5 \\ 1 & x_6 & x_6^2 & x_6^3 & x_6^4 & x_6^5 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix}$$

Notemos que el sistema es de la forma $V^T c = y$ donde V es la matriz de Vandermonde en las variables x_i.

Para resolverlo se utilizaron las funciones `FACTOR/SOLVE` (disponibles en la tarea pasada) y una vez obtenidos los coeficientes $c_1, \cdots, c_6$ de $p(x)$ se evaluó en los x restantes que no se consideraron para la interpolación (valores no sombreados en los datos). Para corroborarlos, se calculó el error relativo con respecto a los valores correctos y también se muestra en el programa.

```
%Introducimos los 6 puntos a interpolar (x_i,y_i)
x=[10;20;40;60;80;100];
y=[1.498;2.138;2.840;2.542;1.877;1.201];
%Generamos la matriz de Vandermonde en las variables x_i
V=fliplr(vander(x));
%Usamos FACTOS/SOLVE para solucionar el sistema V'c=y
[L,U,P,condicion]=FACTOR(V);
[c]=SOLVE(L,U,P,y);
%Mostramos el resultado, los coeficientes c_1,...,c_6
disp('Coeficientes c_i:'); disp(c);
%Mostramos el número de condición del sistema
disp('Número de condición del sistema:'); disp(condicion);

%Ahora evaluamos el polinomio interpolante con Horner,
x_restantes=[5;15;30;50;70;90];
y_restantes=[1.226;1.822;2.662;2.807;2.210;1.539];
y_evaluados=zeros(6,1); errores=zeros(6,1);
%Evaluamos p(x_restante(i)) para los 6 puntos no interpolados
for i=1:6
    res=c(6);
    for j=5:-1:1
        res=res*x_restantes(i)+c(j);
    end
    y_evaluados(i)=res;
    errores(i)=abs(res-y_restantes(i))/y_restantes(i);
end
disp('Polinomio interpolante evaluado en los x restantes:');
disp(y_evaluados);
disp('Errores relativos con respecto a los valores correctos:');
disp(errores);
```

```
Command Window
Coeficientes c_i:
         0.987047619047612
        0.0301762301587316
       0.00300626488095228
     -0.000101749007936505
      1.06930803571426e-06
     -3.80481150793641e-09

Número de condición del sistema:
           153759820426.33

Polinomio interpolante evaluado en los x restantes:
          1.20102319335937
          1.82394220842634
          2.62443229166666
          2.78706696428571
          2.20951116071429
          1.56889583333333

Errores relativos con respecto a los valores correctos:
         0.020372599217477
       0.00106597608470866
        0.0141125876533945
       0.00710118835564167
      0.000221194246927038
fx      0.0194254927442054
```

Se obtuvo que el número de condición del sistema es de $cond(A) = 1.5376 \times 10^{11}$, por lo que **es en extremo mal condicionada**. Esto se veía venir porque estamos utilizando la matriz de Vandermonde, la cual tiene por característica ir aumentando exponencialmente su número de condición conforme se aumenta su dimensión.

El polinomio interpolante calculado por este método es

$$p(x) = 0.98705 + 0.030176x + 0.0030063x^2 - 0.00010175x^3 + 1.0693 \times 10^{-6}x^4 - 3.8048 \times 10^{-9}x^5$$

Se puede pensar que no es confiable por la condición del sistema. Sin embargo, los **errores relativos** de evaluar $p(x)$ en los puntos restantes con respecto a cada valor correcto es **relativamente pequeño** (para tratarse de la matriz de Vandermonde), el más grande con magnitud del orden de 10^{-2} y el más pequeño del orden de 10^{-4}.

Se graficó $p(x)$ y los datos considerados en la Fig. 2.

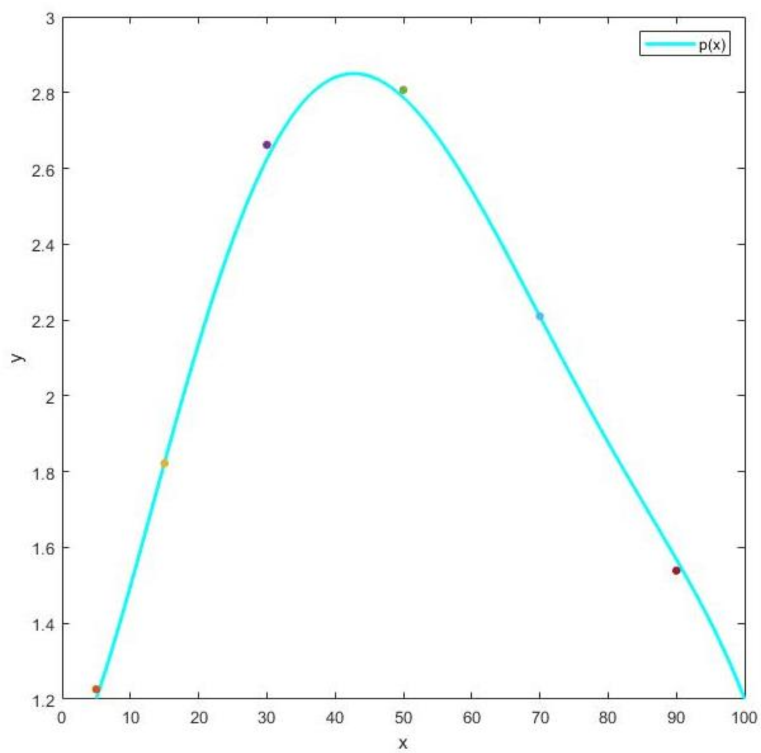

Fig. 2. $p(x)$ obtenido con 6 puntos junto con puntos no considerados para la interpolación.

3.5 There are several ways to evaluate $p_n(x) = c_1 + c_2x + \cdots c_n x^{n-1}$. As a first algorithm we could use

$$
\begin{aligned}
&\mathrm{P: c_1}\\
&\text{for } \mathrm{i} = 2,3,\dots,\mathrm{n} \text{ begin}\\
&\qquad \mathrm{P} \coloneqq \mathrm{P} + \mathrm{c_i} * x^{i-1}\\
&\text{end}
\end{aligned}
$$

How many multiplications does this algorithm require? A better approach is based on the nested form of $p_n(x)$ used in Section 3.3:

$$c1 \ + \ x\{ \ c2 \ + \ x[\ c3 \ + \ x \ (\ c4 \ + \cdots + cNx) \ \cdots]\}$$

The new algorithm is

$$
\begin{aligned}
&\mathrm{P: c_n}\\
&\text{for } \mathrm{i} = \mathrm{n}-1, \mathrm{n}-2, \dots, 1 \text{ begin}\\
&\qquad \mathrm{P} \coloneqq \mathrm{P} * \mathrm{x} + \mathrm{c_i}\\
&\text{end}
\end{aligned}
$$

Compare the number of multiplications for this algorithm to those for the first one.

Para el primer algoritmo tenemos

- x^{i-1} hace $i-1$ multiplicaciones, asumiendo que cada x^i se realiza como $x^i = \prod_{j=1}^{i} x$.
- $c_i * x^{i-1}$ representa entonces i multiplicaciones.
- En la i-ésima iteración, se realizan entonces i multiplicaciones, por lo tanto, el número total es $\sum_{i=2}^{n} i = -1 + \sum_{i=1}^{n} i = \frac{n(n+1)}{2} - 1 \approx \frac{n(n+1)}{2} \approx n^2$

El primer algoritmo hace aproximadamente n^2 multiplicaciones.

Para el segundo algoritmo tenemos

- $P * x$ hace 1 multiplicación.
- En la i-ésima iteración, se realizan solo 1 multiplicación, por lo tanto, el número total es $\sum_{i=1}^{n-1} 1 = n - 1 \approx n$

El segundo algoritmo hace aproximadamente n multiplicaciones.

Claramente el segundo algoritmo ocupa menos multiplicaciones. Este método se conoce como *algoritmo de Horner* y resulta ser muy eficiente, mucho más que el primer algoritmo. Además, tiene la gran ventaja de minimizar el error provocado por las sucesivas multiplicaciones de x, especialmente para potencias grandes, ya que estos resultados pueden ser muy grandes o pequeños.

3.6 Use the algorithm suggested by Exercise 3.4 to compute $p_{12}(x)$, which interpolates all the data in Example 3.5. Plot $p_{12}(x)$ for $5 < x < 100$. Does it look reasonable?

Ahora se pide interpolar con los 12 datos, el doble que en el ejercicio 3.4. Al utilizar el mismo método ahora debemos resolver un sistema lineal de la forma

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{10} & x_1^{11} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{10} & x_2^{11} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_{11} & x_{11}^2 & \cdots & x_{11}^{10} & x_{11}^{12} \\ 1 & x_{12} & x_{12}^2 & \cdots & x_{12}^{10} & x_{12}^{11} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{11} \\ c_{12} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{11} \\ y_{12} \end{pmatrix}$$

Se utilizaron nuevamente `FACTOR/SOLVE` para resolver el sistema y se obtuvieron los coeficientes de $p(x)$. Luego se graficó en el intervalo $5 \leq x \leq 100$ juntos con los 12 puntos, el resultado se muestra en la Fig. 3. Adicionalmente se muestra el número de condición del sistema.

```
%Introducimos los 6 puntos a interpolar (x_i,y_i)
x_r=[5;10;15;20;30;40;50;60;70;80;90;100];
y_r=[1.226;1.498;1.822;2.138;2.662;2.840;...
    2.807;2.542;2.210;1.877;1.539;1.201];
%Generamos la matriz de Vandermonde en las variables x_i
V=fliplr(vander(x_r));
%Usamos FACTOS/SOLVE para solucionar el sistema V'c=y
[L,U,P,condicion]=FACTOR(V);
[c]=SOLVE(L,U,P,y_r);
%Mostramos el resultado, los coeficientes c_1,...,c_12
disp('Coeficientes c_i:'); disp(c);
%Mostramos el número de condición del sistema
disp('Número de condición del sistema:'); disp(condicion);

%Evaluamos y graficamos el polinomio interpolante en [5,100]
x=5:0.1:100; y=zeros(1,length(x));
for i=1:length(x)
    %Evaluación por el algoritmo de Horner
    res=c(12);
    for j=11:-1:1
        res=res*x(i)+c(j);
    end
    y(i)=res;
end
plot(x,y,'g-','linewidth',2); xlabel('x'); ylabel('y'); legend('p(x)');
axis square; axis([0,105,1,3]); hold on;
%Graficamos los puntos interpolados
for i=1:12
    plot(x_r(i),y_r(i),'.','markersize',15,'HandleVisibility','off');
    hold on;
end
```

```
Command Window

>> ejercicio_3_6
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =  8.160034e-25.
> In cond (line 46)
In FACTOR (line 8)
In ejercicio_3_6 (line 8)

Coeficientes c_i:
          2.23349710938026
        -0.596929505356757
         0.130552801505083
       -0.0140406959571093
      0.000917208131186854
     -3.83727490210133e-05
      1.05577547111062e-06
     -1.92804690031042e-08
      2.31070925062083e-10
     -1.74424935439879e-12
      7.51229934430001e-15
     -1.40617895965316e-17

Número de condición del sistema:
      2.03162084773726e+24
```

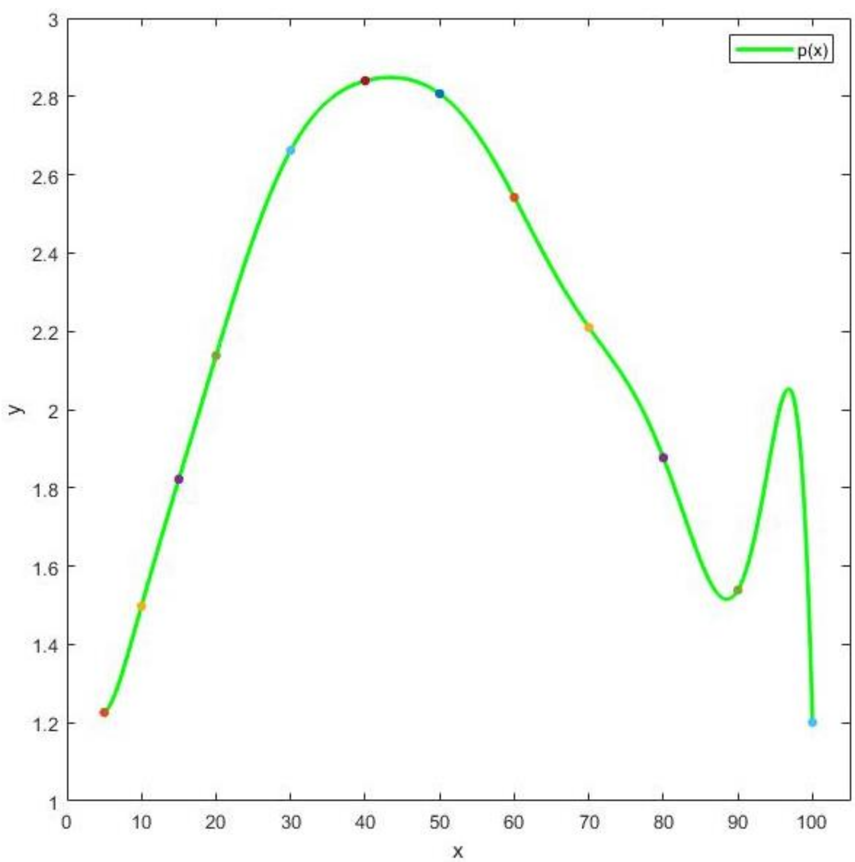

Fig. 3. $p(x)$ junto con puntos utilizados para interpolación.

El número de condición del sistema es 2.03×10^{24}, esto puede significar que los coeficientes obtenidos son poco fiables, sin embargo, el polinomio interpolante concuerda de buena manera con los 12 puntos interpolados.

3.7 Verify the plot in the Figure 3.3 of $w_9(x)$ to see that it has smallest magnitude near the middle of the data. Choose $x_i = -5 + i$ and evaluate $w_9(x)$ at $\{\pm 0.5, \pm 1.5, \cdots, \pm 4.5, \pm 5\}$

El teorema del error de la interpolación incluye el factor $w_N(x) = \prod_{i=1}^{N}(x - x_i)$. Para ver que tan grande se magnifica este factor consideraremos $x_i = -5 + i$ y $N = 9$, entonces

$$w_9(x) = \prod_{i=1}^{9}\big(x - (-5 + i)\big) = \prod_{i=1}^{9}(x + 5 - i) = (x + 4)(x + 3) \cdots (x - 3)(x - 4)$$

La evaluación de este polinomio en el intervalo $[-4,4]$ se muestra en la Figura 3.3 del texto. Se evaluó este polinomio en $\{\pm 0.5, \pm 1.5, \cdots, \pm 4.5, \pm 5\}$ y se muestran sus resultados.

Se puede verificar que al evaluar $w_9(x)$ **en x cercanas a la mitad del intervalo considerado, se tiene un valor menor, lo que implica un error menor.** Más aún, para **x cercanas a un nodo (en este caso a $\{-4, \cdots, 4\}$) el valor de $w_9(x)$ es casi nulo**, podemos asegurar esto por la continuidad, ya que justo en los nodos $w_9(x) = 0$. Finalmente, se verifica que **entre más lejano esté x del centro del intervalo, i.e., acercándose a los bordes o sobrepasándolos, el valor de $w_9(x)$ se dispara.** Esto quiere decir que el error de la interpolación para puntos fuera del intervalo de los nodos es extremadamente grande, por lo que resulta poco conveniente realizar extrapolación. Para un mayor entendimiento, se graficaron los valores en las Figuras 4 y 5.

```
%Introducimos los puntos {-5,-4.5,...,4.5,5}
x=[-5.5:1:5.5]'; x(1)=-5; x(12)=5;
y=zeros(12,1);
%Evaluamos en W_9(x) y graficamos
for k=1:12
    res=1;
    for i=-4:4
        res=res*(x(k)-i);
    end
    y(k)=res;
    plot(x(k),y(k),'o','markersize',7,'markerfacecolor','y',...
        'HandleVisibility','off'); hold on;
end
%Imprimimos los valores obtenidos
disp('Evaluaciones de w_9(x_i) con x_i en {-5,-4.5,...,4.5,5}');
disp([x,y]);
%Graficación de w_9(x) en [-5,5] (OPCIONAL)
x=-5:0.1:5; y=x;
for k=1:length(x)
    res=1;
    for i=-4:4
        res=res*(x(k)-i);
    end
    y(k)=res;
end
plot(x,y,'b--','linewidth',2);
xlabel('x'); ylabel('w_9(x)'); legend('w_9(x)'); axis square;
%Ejes coordenados (OPCIONAL)
plot([-5.5,5.5],[0,0],'k','HandleVisibility','off');
plot([0,0],[-362880,362880],'k','HandleVisibility','off');
```

Command Window

```
>> ejercicio_3_7
Evaluaciones de w_9(x_i) con x_i en {-5,-4.5,...,4.5,5}
                        -5                 -362880
                      -4.5          -67303.564453125
                      -3.5            3959.033203125
                      -2.5            -791.806640625
                      -1.5             304.541015625
                      -0.5            -193.798828125
                       0.5             193.798828125
                       1.5            -304.541015625
                       2.5             791.806640625
                       3.5           -3959.033203125
                       4.5           67303.564453125
                         5                  362880
```

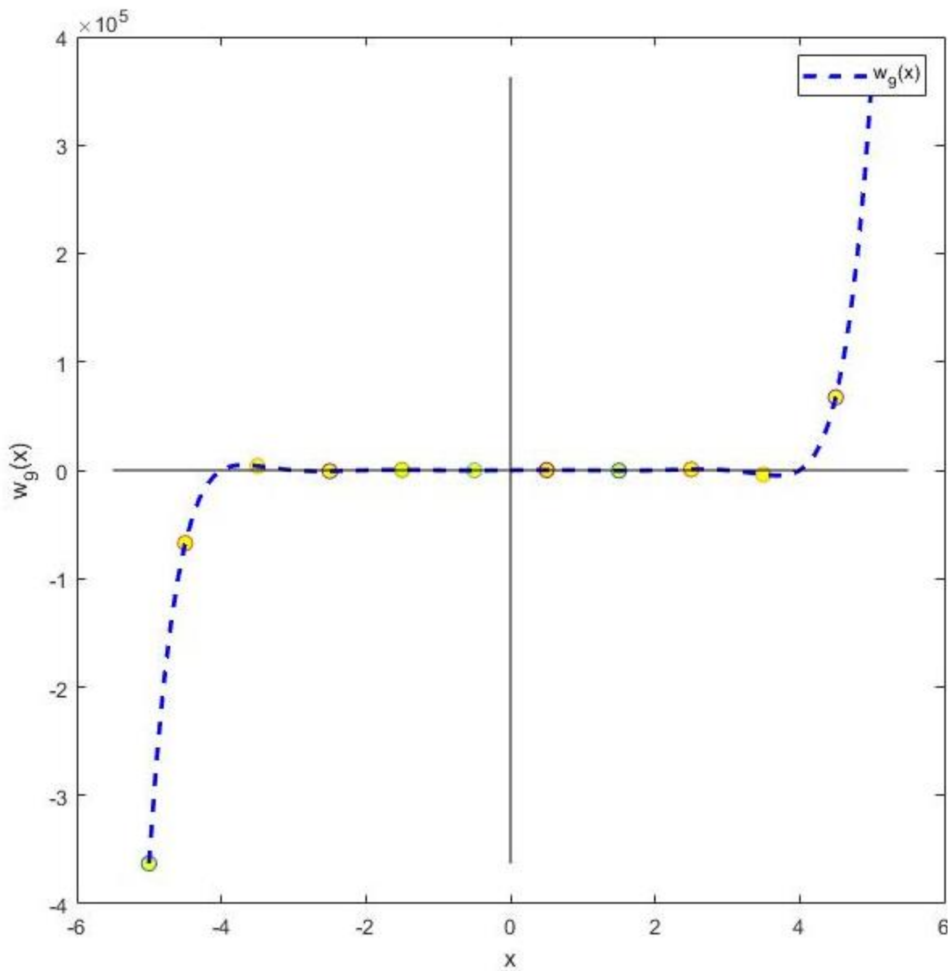

Fig. 4. $w_9(x)$ con $x \in [-5,5]$ con los puntos $(x_i, w_9(x_i))$ con $x_i \in \{-5, -4.5, \cdots, 4.5, 5\}$.

Dado que $w_9 = \prod_{i=-4}^{4}(x - i)$, el intervalo considerado es $[-4,4]$. Se puede verificar que para puntos fuera de $[-4,4]$ los valores de $w_9(x)$ son muy grandes, mientras que para puntos dentro de $[-4,4]$ son relativamente pequeños.

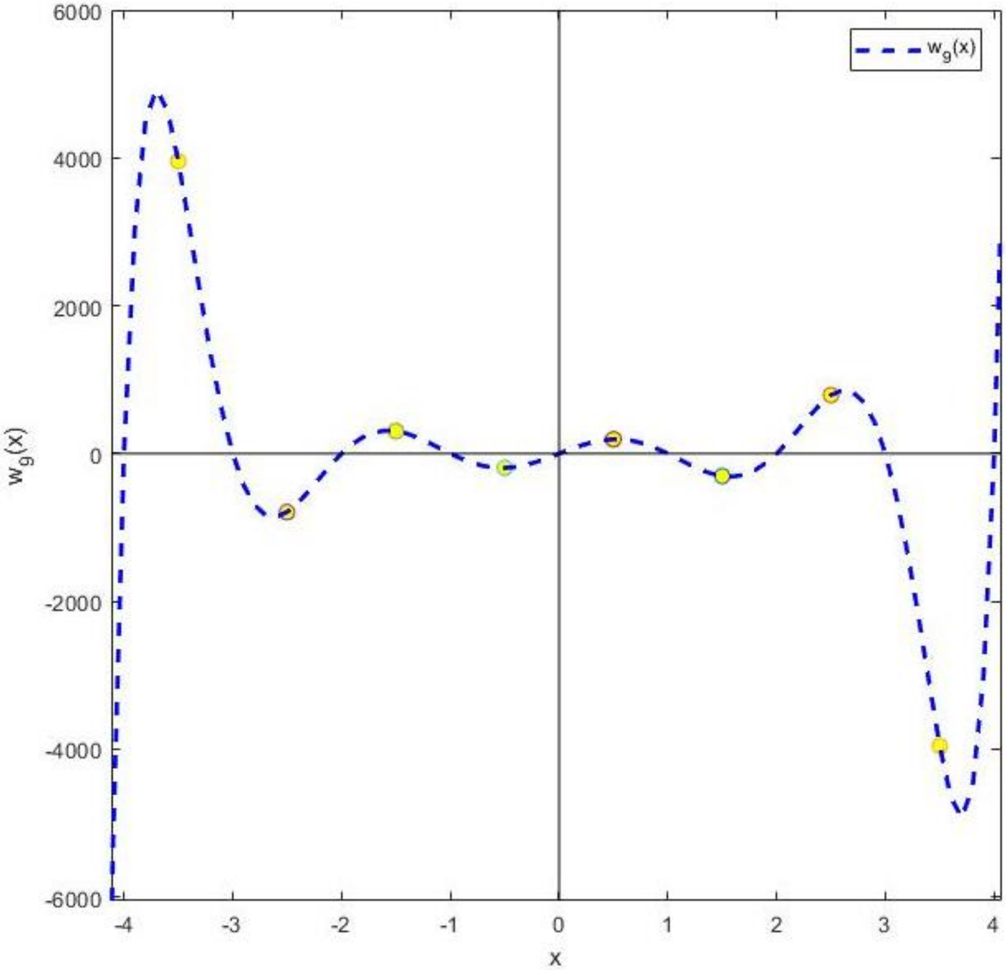

Fig. 5. $w_9(x)$ con $x \in [-4,4]$ con los puntos $(x_i, w_9(x_i))$ con $x_i \in \{-3.5, -2.5 \cdots ,2.5,3.5\}$.

En este caso todos los puntos están dentro del intervalo considerado para $w_9(x)$, el $[-4,4]$. Aunque ahora los valores de $w_9(x)$ son relativamente pequeños en comparación a los analizados en la Fig. 4, resulta evidente que entre más lejos se tome x del centro del intervalo, más grande es $w_9(x)$, asimismo, entre más cerca se tome a x de un nodo, el valor de $w_9(x)$ tiende a ser cero.

3.8 Derivatives of $f(x)$ can be estimated by the corresponding derivative of $P_N(x)$ for some choice of N and $\{x_n\}_{n=1}^{N}$. The usual approach is to try

$$f^{(N-1)}(x) \approx P_N^{(N-1)}(x)$$

Since $P_N(x)$ has degree at most $N-1$, must be a constant function (independent of x).

a) Use $P_N(x) = \sum_{k=1}^{N} f_k L_k(x)$ con $L_k(x) = \prod_{j=1,j\neq k}^{N} \frac{x-x_j}{x_k-x_j}$ to show that

$$P_N^{(N-1)}(x) = (N-1)! \sum_{k=1}^{N} \frac{f_k}{\prod_{j\neq k}(x_k - x_j)}$$

b) What approximation to $f'(x)$ results when $N = 2$?

a) Expresemos $P_N(x)$ como

$$\begin{aligned} P_N(x) = \sum_{k=1}^{N} f_k L_k(x) &= \sum_{k=1}^{N} f_k \prod_{j=1,j\neq k}^{N} \frac{x-x_j}{x_k-x_j} \\ &= \sum_{k=1}^{N} \left(f_k \prod_{j=1,j\neq k}^{N} \frac{1}{x_k - x_j} \right) \left(\prod_{j=1,j\neq k}^{N} x - x_j \right) \\ &= \sum_{k=1}^{N} \left(\frac{f_k}{\prod_{j\neq k}(x_k - x_j)} \right) \left(\prod_{j=1,j\neq k}^{N} x - x_j \right) \end{aligned}$$

Denotemos por c_k el término $\left(\frac{f_k}{\prod_{j\neq k}(x_k - x_j)}\right)$. Claramente c_k es una constante pues no depende de la variable x. Tomemos la derivada $(n-1)$ de $P_N(x)$ con respecto a x, según la expresión anterior se obtiene

$$P_N^{(N-1)}(x) = \frac{d^{(N-1)}}{dx^{(N-1)}} \sum_{k=1}^{N} (c_k) \left(\prod_{j=1,j\neq k}^{N} x - x_j \right) = \sum_{k=1}^{N} (c_k) \frac{d^{(N-1)}}{dx^{(N-1)}} \left(\prod_{j=1,j\neq k}^{N} x - x_j \right)$$

Resolvamos $\frac{d^{(N-1)}}{dx^{(N-1)}} \left(\prod_{j=1,j\neq k}^{N} x - x_j \right)$. Notemos que este es un polinomio mónico en la variable x de grado $n-1$ ya que

$$\begin{aligned} \prod_{j=1,j\neq k}^{N} x - x_j &= (x - x_1) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_N) \\ &= x^{N-1} + C_{N-2}x^{N-2} + \cdots + C_1 x + C_0 \end{aligned}$$

Con $C_{N-2}, \cdots, C_0$ coeficientes constantes a determinar. Entonces al tomar la derivada tendremos que ésta vale 0 para los términos con potencias menores que $N-1$ y para el término x^{N-1} se tiene $\frac{d^{(N-1)}}{dx^{(N-1)}} x^{(N-1)} = (N-1)!$. Expresemos estos hechos como:

$$\frac{d^{(N-1)}}{dx^{(N-1)}} \prod_{j=1, j\neq k}^{N} x - x_j = \frac{d^{(N-1)}}{dx^{(N-1)}} (x^{N-1} + C_{N-2}x^{N-2} + \cdots + C_1 x + C_0) = (N-1)!$$

Así, sustituyendo en nuestra expresión $P_N^{(N-1)}(x)$ tendremos que

$$P_N^{(N-1)}(x) = \sum_{k=1}^{N} (c_k) \frac{d^{(N-1)}}{dx^{(N-1)}} \left(\prod_{j=1, j\neq k}^{N} x - x_j \right) = \sum_{k=1}^{N} (c_k)(N-1)! = (N-1)! \sum_{k=1}^{N} (c_k)$$

Recordemos que $c_k = \left(\frac{f_k}{\prod_{j\neq k}(x_k - x_j)} \right)$. Por lo tanto, concluimos que

$$P_N^{(N-1)}(x) = (N-1)! \sum_{k=1}^{N} \frac{f_k}{\prod_{j\neq k}(x_k - x_j)}$$

Que es lo que se quería demostrar■

b) Cuando $N = 2$, utilizando la aproximación $f'(x) \approx P_N'(x)$ tenemos que

$$f'(x) \approx (2-1)! \sum_{k=1}^{2} \frac{f_k}{\prod_{j\neq k}(x_k - x_j)} = \frac{f_1}{(x_1 - x_2)} + \frac{f_2}{(x_2 - x_1)} = \frac{f_1 - f_2}{x_1 - x_2}$$

Para interpretar este resultado, recordemos que los puntos que tenemos de la función $f(x)$ son (x_1, f_1) y (x_2, f_2) en el plano, por lo que esta aproximación a la derivada de f es la pendiente de la recta que une a (x_1, f_1) con (x_2, f_2).
Esta aproximación a $f'(x)$ puede ser todavía más exacta si consideramos más de dos puntos y cada vez más juntos, ya que así nos acercaríamos más a obtener la pendiente de la recta tangente a f.

3.10 Deriva las ecuaciones (tomando $h = x_{n+1} - x_n$)

$$a = f_n, \quad b = f_n', \qquad c = \frac{\frac{3(f_{n+1} - f_n)}{h} - 2f_n' - f_{n+1}'}{h}, \qquad d = \frac{f_n' + f_{n+1}' - \frac{2(f_{n+1} - f_n)}{h}}{h^2}$$

Comienza con que $H(x) = a + b(x - x_n) + c(x - x_n)^2 + d(x - x_n)^3$ y después deriva y resuelve las cuatro ecuaciones para a, b, c y d resultado de las condiciones

$$H(x_n) = f_n, \qquad H(x_{n+1}) = f_{n+1}, \qquad H'(x_n) = f_n', \qquad H'(x_{n+1}) = f_{n+1}'$$

Comencemos por escribir

$$H(x) = a + b(x - x_n) + c(x - x_n)^2 + d(x - x_n)^3$$
$$H'(x) = b + 2c(x - x_n) + 3d(x - x_n)^2$$

- Tomemos la condición $H(x_n) = f_n$, entonces

$$H(x_n) = a + b(x_n - x_n) + c(x_n - x_n)^2 + d(x_n - x_n)^3 = a = f_n$$

  Por lo tanto

$$\boldsymbol{a = f_n}.$$

- Tomemos la condición $H'(x_n) = f_n'$, entonces

$$H'(x_n) = b + 2c(x_n - x_n) + 3d(x_n - x_n)^2 = b = f_n'$$

  Por lo tanto

$$\boldsymbol{b = f_n'}.$$

- Tomemos la condición $H(x_{n+1}) = f_{n+1}$ y denotemos $h = x_{n+1} - x_n$, entonces

$$H(x_{n+1}) = a + b(x_{n+1} - x_n) + c(x_{n+1} - x_n)^2 + d(x_{n+1} - x_n)^3 = a + bh + ch^2 + dh^3$$
$$= f_n + f_n'h + ch^2 + dh^3 = f_{n+1}$$

  Utilizando los resultados anteriores $a = f_n$ y $b = f_n'$. Entonces tenemos la ecuación

$$ch^2 + dh^3 = f_{n+1} - f_n - hf_n'$$

  O bien

$$c + dh = \frac{f_{n+1} - f_n - hf_n'}{h^2} \tag{1}$$

- Tomemos la condición $H'(x_{n+1}) = f_{n+1}'$, entonces

$$H'(x_{n+1}) = b + 2c(x_{n+1} - x_n) + 3d(x_{n+1} - x_n)^2 = b + 2ch + 3dh^2$$
$$= f_n' + 2ch + 3dh^2 = f_{n+1}'$$

Utilizando el resultado anterior $b = f_n'$. Entonces tenemos la ecuación

$$2ch + 3dh^2 = f_{n+1}' - f_n'$$

O bien

$$2c + 3dh = \frac{f_{n+1}' - f_n'}{h} \tag{2}$$

Despejando c de la ecuación (1) y sustituyendo en (2) tenemos que

$$2c + 3dh = 2\left(\frac{f_{n+1} - f_n - hf_n'}{h^2} - dh\right) + 3dh = \frac{f_{n+1}' - f_n'}{h}$$

$$\Rightarrow dh = \frac{f_{n+1}' - f_n'}{h} - 2\left(\frac{f_{n+1} - f_n - hf_n'}{h^2}\right) = \frac{h(f_{n+1}' - f_n') - 2f_{n+1} + 2f_n + 2hf_n'}{h^2}$$

$$= \frac{hf_{n+1}' + hf_n' - 2(f_{n+1} - f_n)}{h^2} = \frac{h\left(f_{n+1}' + f_n' - \frac{2(f_{n+1} - f_n)}{h}\right)}{h^2}$$

$$= \frac{f_{n+1}' + f_n' - \frac{2(f_{n+1} - f_n)}{h}}{h} = dh$$

Por lo tanto

$$\boldsymbol{d = \frac{f_{n+1}' + f_n' - \frac{2(f_{n+1} - f_n)}{h}}{h^2}}$$

Finalmente, sustituyendo este valor en la ecuación (1) se tiene que

$$\mathrm{c} + dh = c + \left(\frac{f_{n+1}' + f_n' - \frac{2(f_{n+1} - f_n)}{h}}{h^2}\right)\mathrm{h} = \frac{f_{n+1} - f_n - hf_n'}{h^2}$$

$$\Rightarrow c = \frac{f_{n+1} - f_n - hf_n'}{h^2} - \frac{f_{n+1}' + f_n' - \frac{2(f_{n+1} - f_n)}{h}}{h}$$

$$= \frac{\frac{f_{n+1} - f_n - hf_n'}{h} - f_{n+1}' - f_n' + \frac{2(f_{n+1} - f_n)}{h}}{h}$$

$$= \frac{\frac{f_{n+1} - f_n - hf_n' + 2f_{n+1} - 2f_n}{h} - f_n' - f_{n+1}'}{h}$$

$$= \frac{\frac{3f_{n+1} - 3f_n}{h} - f_n' - f_n' - f_{n+1}'}{h} = \frac{\frac{3(f_{n+1} - f_n)}{h} - 2f_n' - f_{n+1}'}{h}$$

Por lo tanto

$$\boldsymbol{c = \frac{\frac{3(f_{n+1} - f_n)}{h} - 2f_n' - f_{n+1}'}{h}}$$