3.11 Verify that using polynomials interpolating at the $N = 2m + 1$ equally spaced points $x_j = -5 + \frac{5(j-1)}{m}$ give poor approximations to Runge's function $f(x) = \frac{1}{1+x^2}$ on $[-5,5]$.

a) Compute the maximum value of $|f(x) - p_{2m+1}(x)|$ over a large set of x values (not interpolating points) in $[-5,5]$ for $m = 7$, $m = 10$, and $m = 13$. Are the errors increasing or decreasing as m gets bigger?

b) Repeat (a) but this time only compute error on $[-1,1]$. Use the same $\{x_j\}$ and the same three m values as in (a). What happens this time as N increases?

Para calcular el polinomio interpolante se utilizará la forma de Lagrange, la solución del sistema con la matriz de Vandermonde e interpolación de Hermite vía la función `pchip` de Matlab.

Se escribió un script en Matlab para que, dados los valores m (para obtener $2m + 1$ puntos para interpolar en $[-5,5]$), a y b (límites del intervalo $[a,b]$ que es dividido uniformemente para la consideración del error absoluto al interpolar), se obtenga el máximo valor absoluto $|f(x) - p_{2m+1}(x)|$ para x en $\left\{x_i = a + \frac{b-a}{100} i : i = 0,1,\cdots,100\right\}$, es decir, hemos subdividido $[a,b]$ en 100 valores uniformemente espaciados.

a) En este caso se corrió varias veces el programa con los distintos valores de $m = 7,10,13$ y eligiendo $a = -5$, $b = 5$. Se presentan los resultados obtenidos en la Tabla 1 y la graficación de los polinomios interpolantes para $m = 7$ en la Figura 1.

Tabla 1. Máximos errores absolutos para distintos métodos de interpolación con puntos uniformemente espaciados considerados en el intervalo $[-5,5]$ para la función de Runge.

Forma de $p_{2m+1}(x)$	Máximo valor absoluto para:		
	$m = 7$ (15 puntos)	$m = 10$ (21 puntos)	$m = 13$ (27 puntos)
Lagrange	7.192	58.278	535.07
Vandermonde	7.192	58.278	535.07
Hermite	0.016383	0.012471	0.0079798

Entre más crece el número de puntos considerados a interpolar, más **crece** la magnitud del error absoluto máximo para la forma de Lagrange y de Vandermonde que curiosamente resultan ser iguales. Sin embargo, sucede lo contrario con Hermite pues a más puntos considerados, **menor** es la magnitud máxima del error absoluto, es decir, aproxima mejor la función.

b) Ahora se corrió varias veces el programa con los distintos valores de $m = 7,10,13$ pero eligiendo $a = -1$, $b = 1$. Se presentan los resultados obtenidos en la Tabla 2 y la graficación de los polinomios interpolantes para $m = 7$ en la Figura 2.

Tabla 2. Máximos errores absolutos para distintos métodos de interpolación con puntos uniformemente espaciados considerados en $[-1,1]$ para la función de Runge.

Forma de $p_{2m+1}(x)$	Máximo valor absoluto para:		
	$m = 7$ (15 puntos)	$m = 10$ (21 puntos)	$m = 13$ (27 puntos)
Lagrange	0.019716	0.0032773	0.00074755
Vandermonde	0.019716	0.0032773	0.00074759
Hermite	0.016975	0.012625	0.0087972

Entre más crece el número de puntos considerados a interpolar, más **decrece** la magnitud del error absoluto máximo para la forma de Lagrange y de Vandermonde que curiosamente resultan ser iguales (otra vez). Sin embargo, con Hermite también **decrece** la magnitud máxima del error absoluto, pero llega un momento en que **el error absoluto es mayor que Lagrange y Vandermonde**, algo que no había pasado anteriormente.

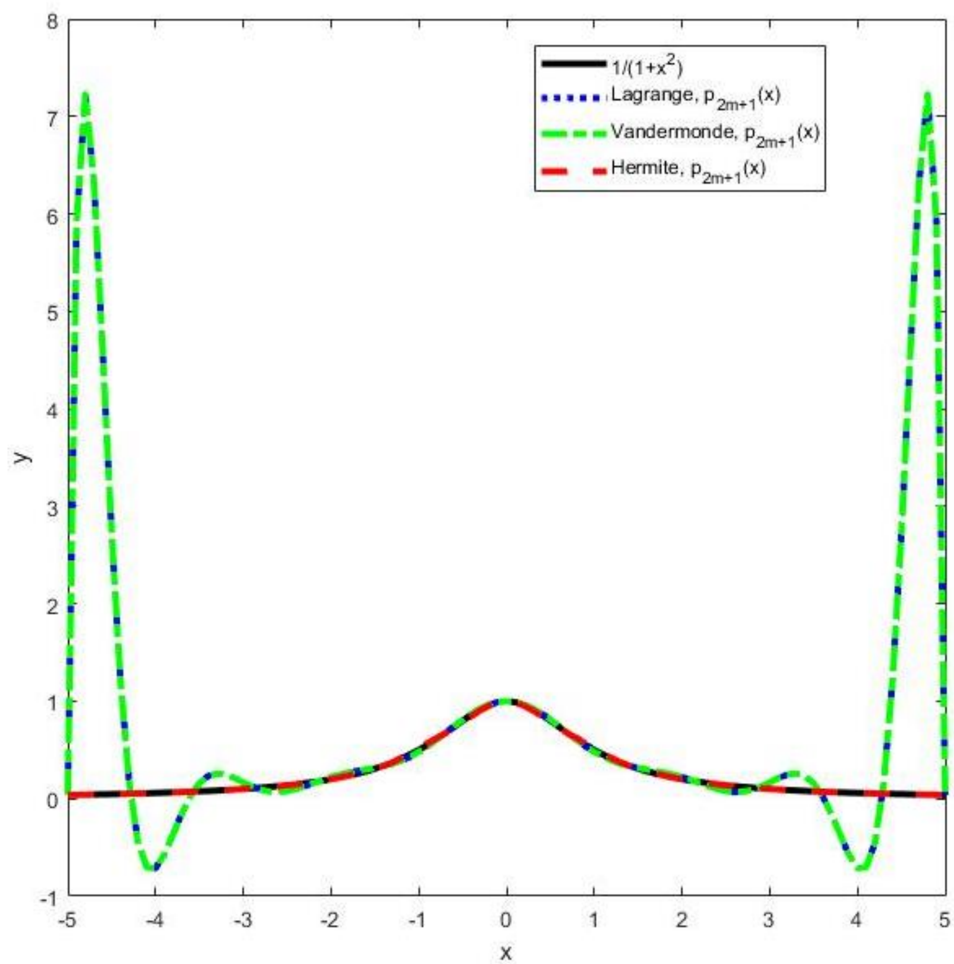

Figura 1. Función de Runge en [-5,5] y polinomios interpolantes en la forma de Lagrange, Vandermonde y Hermite con puntos interpolados en [-5,5].

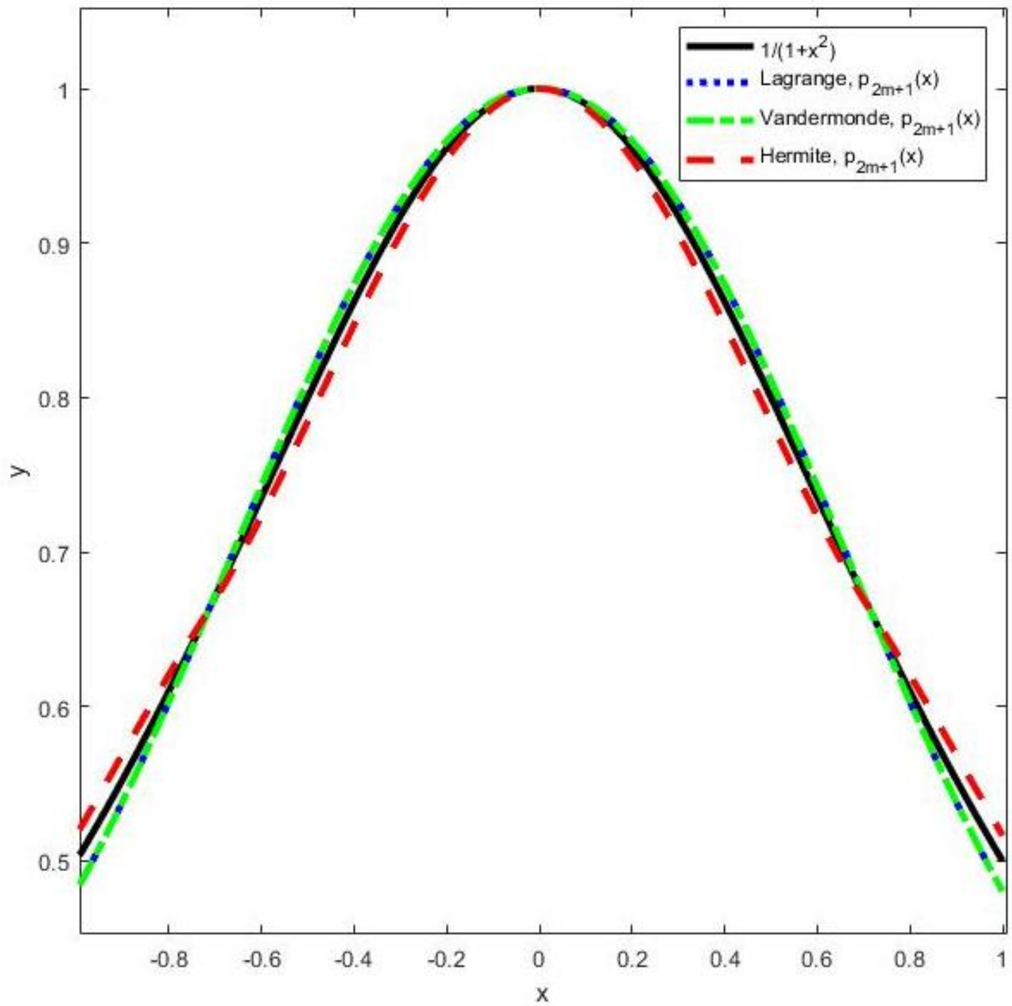

Figura 2. Función de Runge en [-1,1] y polinomios interpolantes en la forma de Lagrange, Vandermonde y Hermite con puntos interpolados en [-5,5].

```
%Numero m para generar n=2m+1 puntos, extremos del intervalo [a,b]
m=13; a=-1; b=1;
%Generamos nuestros puntos para interpolar en la f. de Runge
x_m=[-5:5/m:5]'; y_m=1./(1.+x_m.^2);
%Generamos nuestro conjunto de grande de puntos en [-5,5] en la f.
Runge
x=[a:(b-a)/100:b]'; y_runge=1./(1.+x.^2);
%INTERPOLACIÓN DE LAGRANGE PARA LOS 2m+1 PUNTOS
error=0;
for k=1:length(x)
    y=0;
    %Evaluando en el polinomio de Lagrange p_2m+1(x)
    for i=1:length(y_m)
        termino=y_m(i);
        for j=1:length(x_m)
            if i~=j
```

```
                termino=termino*((x(k)-x_m(j))/(x_m(i)-x_m(j)));
            end
        end
        y=y+termino;
    end
    %Actualizamos el error más grande si es que lo es
    error_absoluto=abs(y_runge(k)-y);
    if error_absoluto>error
        error=error_absoluto;
    end
end
disp('Máximo error absoluto de p(x)_2m+1 con Lagrange:');
disp(error);

%INTERPOLACIÓN DE VANDERMONDE PARA LOS 2m+1 PUNTOS
%Generamos la matriz de Vandermonde para los puntos interpolados
A=fliplr(vander(x_m));
%Resolvemos el sistema Ac=y_m para obtener los coeficientes de
p_2m+1(x)
[L,U,P]=lu(A); aux=L\(P*y_m); c=U\aux; c=flip(c);
%Los coeficientes de p_2m+1(x) están en c, computamos el error
error=0;
for i=1:length(x)
    y=polyval(c,x(i));
    error_absoluto=abs(y-y_runge(i));
    if error_absoluto>error
        error=error_absoluto;
    end
end
disp('Máximo error absoluto de p(x)_2m+1 con Vandermonde:');
disp(error);

%INTERPOLACIÓN CÚBICA DE HERMITE PARA LOS 2m+1 PUNTOS
%Usamos la función pchip para interpolar (x_m,y_m)
c=pchip(x_m,y_m);
%Actualmente en c están los coeficientes del polinomio interpolante
error=0;
for i=1:length(x)
    y=ppval(c,x(i));
    error_absoluto=abs(y-y_runge(i));
    if error_absoluto>error
        error=error_absoluto;
    end
end
disp('Máximo error absoluto de p(x)_2m+1 con Hermite:');
disp(error);
```

3.12 Verify that using polynomials interpolating at the Chebyshev points gives good approximations to Runge's function. As in the preceding exercise, compute the maximum value of $|f(x) - P_N(x)|$ over a large set of x values (not interpolating points) in $[-5,5]$ for $N = 15$, $N = 21$, and $N = 27$. What is the behavior of the errors as N gets bigger?

En este ejercicio simplemente se modificó el programa anterior para cambiar la generación de los puntos interpolantes $\left\{\left(x_j, f\left(x_j\right)\right)\right\}_{j=1}^{N}$ con $f(x) = \frac{1}{1+x^2}$. En lugar de tener puntos uniformemente espaciados ahora corresponden con los puntos de Chebysev en el intervalo $[a, b]$, donde $a = -5$ y $b = 5$ dados por

$$x_j = \frac{b+a}{2} + \frac{b-a}{2}\cos\frac{(2j-1)\pi}{2N}, j = 1,2,\cdots, N$$

Se muestran los distintos errores absolutos máximos obtenidos para cada forma del polinomio interpolante en la Tabla 3 y se grafican para $N = 15$ en la Figura 3.

Tabla 3. Máximos errores absolutos para distintos métodos de interpolación con puntos de Chebysev considerados en el intervalo $[-5,5]$ para la función de Runge.

Forma de $p_{2m+1}(x)$	Máximo valor absoluto para:		
	$N = 15$ puntos	$N = 21$ puntos	$N = 27$ puntos
Lagrange	0.046517	0.015325	0.004583
Vandermonde	0.046517	0.015325	0.004583
Hermite	0.016231	0.016829	0.014343

La diferencia es abismal comparando con el caso de puntos uniformemente espaciados. Aquí, entre más puntos se tengan menor es la magnitud del error absoluto máximo en cualquier forma del polinomio interpolante con excepción de la forma de Hermite, donde pareciera que el error oscila, pero a una amplitud considerablemente pequeña. Como comentario adicional los errores de las formas de Lagrange y Vandermonde coinciden nuevamente.

```
%Numero m de puntos Chebysev
m=27; a=-5; b=5;
%Generamos nuestros puntos para interpolar
x_m=zeros(m,1);
for j=1:m
    %Raíces del polinomio de Chebysev
    x_m(j)=(b+a)/2 + ((b-a)/2)*cos((2*j-1)*pi/(2*m));
end
y_m=1./(1.+x_m.^2);
%Generamos nuestro conjunto de grande de puntos en [-5,5] en la f.
Runge
```

```
x=[a:(b-a)/100:b]'; y_runge=1./(1.+x.^2);

%INTERPOLACIÓN DE LAGRANGE PARA LOS 2m+1 PUNTOS
error=0;
for k=1:length(x)
    y=0;
    for i=1:length(y_m)
        termino=y_m(i);
        for j=1:length(x_m)
            if i~=j
                termino=termino*((x(k)-x_m(j))/(x_m(i)-x_m(j)));
            end
        end
        y=y+termino;
    end
    error_absoluto=abs(y_runge(k)-y);
    if error_absoluto>error
        error=error_absoluto;
    end
end
disp('Máximo error absoluto de p(x)_2m+1 con Lagrange:');
disp(error);

%INTERPOLACIÓN DE VANDERMONDE PARA LOS 2m+1 PUNTOS
A=fliplr(vander(x_m));
[L,U,P]=lu(A); aux=L\(P*y_m); c=U\aux; c=flip(c);
error=0;
for i=1:length(x)
    y=polyval(c,x(i));
    error_absoluto=abs(y-y_runge(i));
    if error_absoluto>error
        error=error_absoluto;
    end
end
disp('Máximo error absoluto de p(x)_2m+1 con Vandermonde:');
disp(error);

%INTERPOLACIÓN CÚBICA DE HERMITE PARA LOS 2m+1 PUNTOS
c=pchip(x_m,y_m);
error=0;
for i=1:length(x)
    y=ppval(c,x(i));
    error_absoluto=abs(y-y_runge(i));
    if error_absoluto>error
        error=error_absoluto;
    end
end
disp('Máximo error absoluto de p(x)_2m+1 con Hermite:');
disp(error);
```

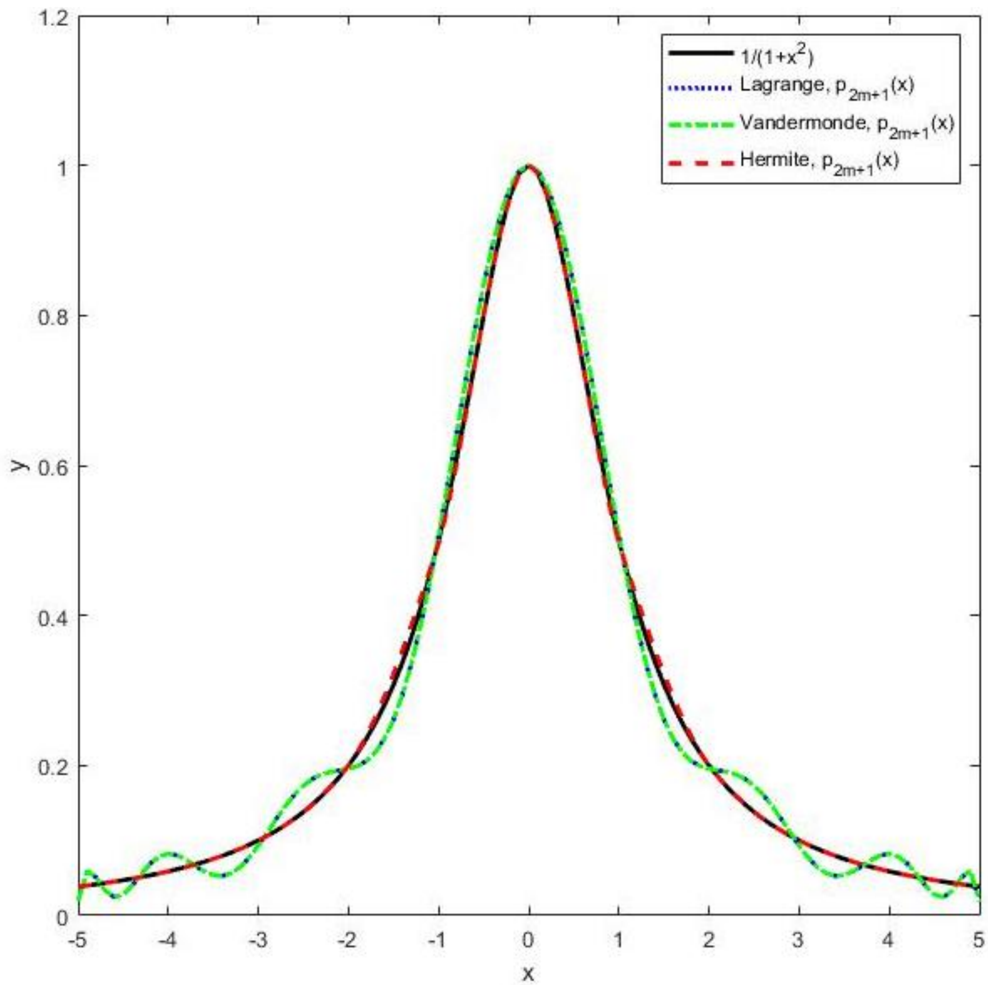

Figura 3. Función de Runge junto a polinomios interpolantes tomando los puntos de Chebysev en el intervalo [-5,5].

3.13 Repeat Exercise 3.11b for the function $f(x) = |x|$ on $[-1,1]$. The $\{x_j\}$ are now $x_j = -1 + (j-1)/m$ for $j = 1, 2, \cdots, 2m+1$.

Simplemente se tomó la rutina del ejercicio 3.11 y se cambió la función al valor absoluto $f(x) = |x|$. Los máximos errores absolutos obtenidos se muestran en la Tabla 4 y se grafica el resultado para $m = 7$ en la Figura 4.

Tabla 4. Máximos errores absolutos para distintos métodos de interpolación con puntos uniformemente espaciados considerados en el intervalo $[-1,1]$ para el valor absoluto.

Forma de $p_{2m+1}(x)$	Máximo valor absoluto para:		
	$m = 7$ (15 puntos)	$m = 10$ (21 puntos)	$m = 13$ (27 puntos)
Lagrange	4.0609	92.699	2916.6
Vandermonde	4.0609	92.699	2916.6
Hermite	0.020736	0.0144	0.010952

Sucede un efecto similar al fenómeno de Runge con espaciamiento uniforme, entre más puntos consideremos el máximo error absoluto incrementa considerablemente en la forma de Lagrange y Vandermonde. Por otro lado, con Hermite tenemos mejoras entre más puntos consideremos, un comportamiento que se ha manifestado en los otros experimentos.

```
%Numero m para generar n=2m+1 puntos, extremos del intervalo [a,b]
m=7; a=-1; b=1;
%Generamos nuestros puntos para interpolar f(x)=abs(x)
x_m=[-1:1/m:1]'; y_m=abs(x_m);
%Generamos nuestro conjunto de grande de puntos en [1,1]
x=[a:(b-a)/100:b]'; y_abs=abs(x);
%INTERPOLACIÓN DE LAGRANGE PARA LOS 2m+1 PUNTOS
error=0;
for k=1:length(x)
    y=0;
    %Evaluando en el polinomio de Lagrange p_2m+1(x)
    for i=1:length(y_m)
        termino=y_m(i);
        for j=1:length(x_m)
            if i~=j
                termino=termino*((x(k)-x_m(j))/(x_m(i)-x_m(j)));
            end
        end
        y=y+termino;
    end
    %Actualizamos el error más grande si es que lo es
    error_absoluto=abs(y_abs(k)-y);
    if error_absoluto>error
        error=error_absoluto;
    end
```

```
end
disp('Máximo error absoluto de p(x)_2m+1 con Lagrange:');
disp(error);

%INTERPOLACIÓN DE VANDERMONDE PARA LOS 2m+1 PUNTOS
%Generamos la matriz de Vandermonde para los puntos interpolados
A=fliplr(vander(x_m));
%Resolvemos el sistema Ac=y_m para obtener los coeficientes de
p_2m+1(x)
[L,U,P]=lu(A); aux=L\(P*y_m); c=U\aux; c=flip(c);
%Los coeficientes de p_2m+1(x) están en c, computamos el error
error=0;
for i=1:length(x)
    y=polyval(c,x(i));
    error_absoluto=abs(y-y_abs(i));
    if error_absoluto>error
        error=error_absoluto;
    end
end
disp('Máximo error absoluto de p(x)_2m+1 con Vandermonde:');
disp(error);

%INTERPOLACIÓN CÚBICA DE HERMITE PARA LOS 2m+1 PUNTOS
%Usamos la función pchip para interpolar (x_m,y_m)
c=pchip(x_m,y_m);
%Actualmente en c están los coeficientes del polinomio interpolante
error=0;
for i=1:length(x)
    y=ppval(c,x(i));
    error_absoluto=abs(y-y_abs(i));
    if error_absoluto>error
        error=error_absoluto;
    end
end
disp('Máximo error absoluto de p(x)_2m+1 con Hermite:');
disp(error);
```

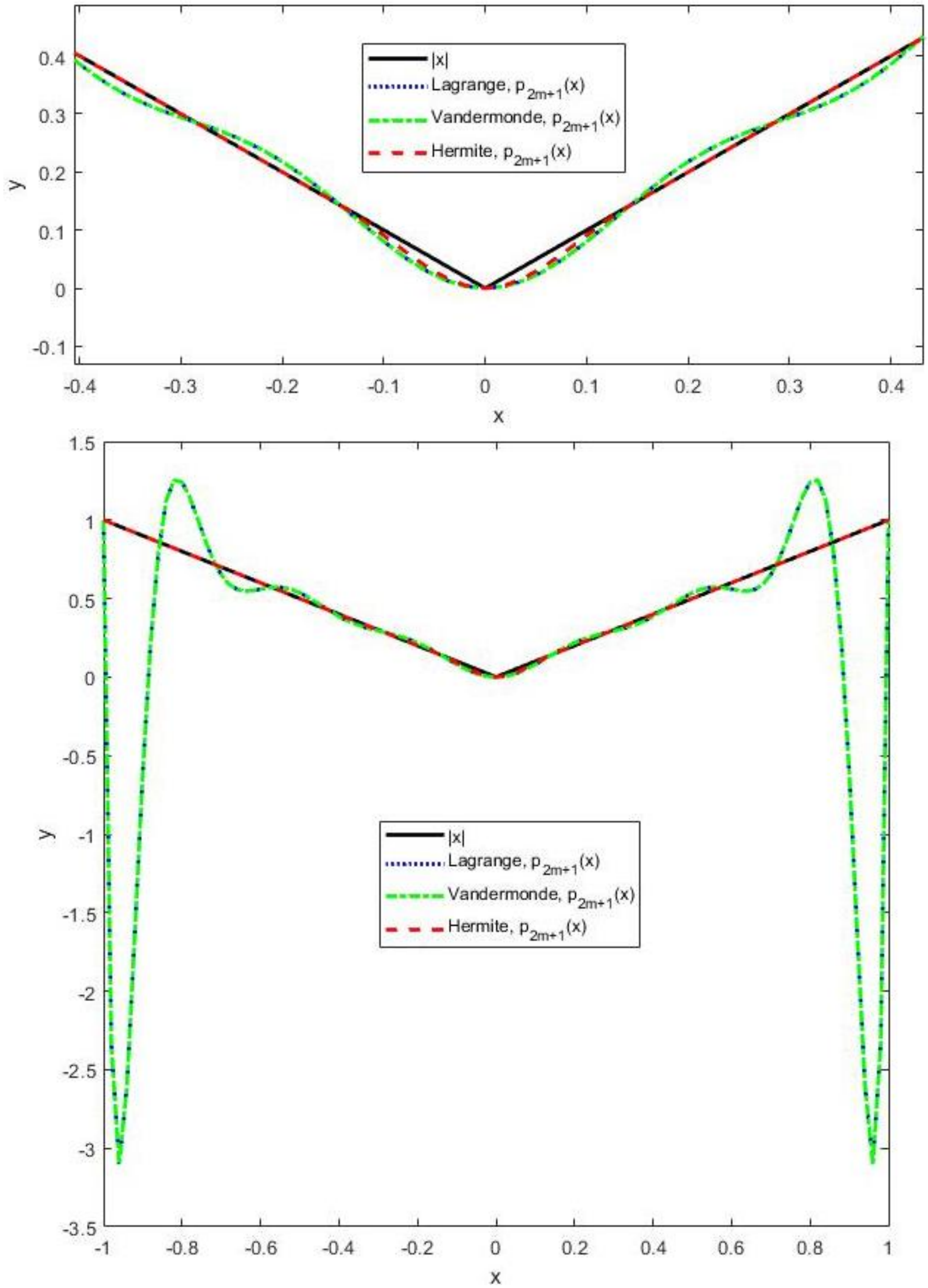

Figura 4. Función valor absoluto junto a polinomios interpolantes en las formas de Lagrange, Vandermonde y Hermite.
En la parte superior se muestra en detalle el intervalo [-0.4,0.4] donde el error absoluto es pequeño. En la parte inferior se muestra el intervalo [-1,1] donde el error absoluto es mucho mayor.

3.16 For the data, calculate $P_5(x)$

x	-2	-1	0	1	2
f	-4	1	1	2	10

a) In the Lagrange form.
b) Using the matrix method discussed in Exercise 3.4 (the linear system is small enough to be solved by hand).
c) In the Newton divided difference form

a) Recordemos $p_5(x) = \sum_{i=1}^{5} y_i L_i(x)$ con $L_i(x) = \prod_{j=1, j\neq i}^{5} \frac{x-x_j}{x_i-x_j}$.
Calculando los respectivos polinomios de Lagrange

$$L_1(x) = \left(\frac{x-(-1)}{-2-(-1)}\right)\left(\frac{x-0}{-2-0}\right)\left(\frac{x-1}{-2-1}\right)\left(\frac{x-2}{-2-2}\right) = \left(\frac{x+1}{-1}\right)\left(\frac{x}{-2}\right)\left(\frac{x-1}{-3}\right)\left(\frac{x-2}{-4}\right)$$

$$\therefore L_1(x) = \frac{1}{24}(x^4 - 2x^3 - x^2 + 2x)$$

$$L_2(x) = \left(\frac{x-(-2)}{-1-(-2)}\right)\left(\frac{x-0}{-1-0}\right)\left(\frac{x-1}{-1-1}\right)\left(\frac{x-2}{-1-2}\right) = \left(\frac{x+2}{1}\right)\left(\frac{x}{-1}\right)\left(\frac{x-1}{-2}\right)\left(\frac{x-2}{-3}\right)$$

$$\therefore L_2(x) = -\frac{1}{6}(x^4 - x^3 - 4x^2 + 4x)$$

$$L_3(x) = \left(\frac{x-(-2)}{0-(-2)}\right)\left(\frac{x-(-1)}{0-(-1)}\right)\left(\frac{x-1}{0-1}\right)\left(\frac{x-2}{0-2}\right) = \left(\frac{x+2}{2}\right)\left(\frac{x+1}{1}\right)\left(\frac{x-1}{-1}\right)\left(\frac{x-2}{-2}\right)$$

$$\therefore L_3(x) = \frac{1}{4}(x^4 - 5x^2 + 4)$$

$$L_4(x) = \left(\frac{x-(-2)}{1-(-2)}\right)\left(\frac{x-(-1)}{1-(-1)}\right)\left(\frac{x-0}{1-0}\right)\left(\frac{x-2}{1-2}\right) = \left(\frac{x+2}{3}\right)\left(\frac{x+1}{2}\right)\left(\frac{x}{1}\right)\left(\frac{x-2}{-1}\right)$$

$$\therefore L_4(x) = -\frac{1}{6}(x^4 + x^3 - 4x^2 - 4x)$$

$$L_5(x) = \left(\frac{x-(-2)}{2-(-2)}\right)\left(\frac{x-(-1)}{2-(-1)}\right)\left(\frac{x-0}{2-0}\right)\left(\frac{x-1}{2-1}\right) = \left(\frac{x+2}{4}\right)\left(\frac{x+1}{3}\right)\left(\frac{x}{2}\right)\left(\frac{x-1}{1}\right)$$

$$\therefore L_5(x) = \frac{1}{24}(x^4 + 2x^3 - x^2 - 2x)$$

$$p(x) = \frac{-4}{24}(x^4 - 2x^3 - x^2 + 2x) + \frac{-1}{6}(x^4 - x^3 - 4x^2 + 4x) + \frac{1}{4}(x^4 - 5x^2 + 4)$$
$$+ \frac{-2}{6}(x^4 + x^3 - 4x^2 - 4x) + \frac{10}{24}(x^4 + 2x^3 - x^2 - 2x)$$
$$= \frac{1}{24}(24x^3 + 12x^2 - 12x + 24) = x^3 + \frac{1}{2}x^2 - \frac{1}{2}x + 1$$

$$\therefore \boldsymbol{p(x) = x^3 + \frac{1}{2}x^2 - \frac{1}{2}x + 1} \blacksquare$$

b) Por este método obtenemos los coeficientes del polinomio interpolante a través del sistema lineal con la matriz de Vandermonde

$$\begin{pmatrix} 1 & -2 & -2^2 & -2^3 & -2^4 \\ 1 & -1 & -1^2 & -1^3 & -1^4 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1^2 & 1^3 & 1^4 \\ 1 & 2 & 2^2 & 2^3 & 2^4 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} = \begin{pmatrix} -4 \\ 1 \\ 1 \\ 2 \\ 10 \end{pmatrix}$$

Aplicando eliminación con pivoteo parcial para factorizar LU obtenemos

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1/2 & 1 & 0 & 0 \\ 1 & 3/4 & 3/4 & 1 & 0 \\ 1 & 1/4 & 3/4 & -1 & 1 \end{pmatrix} \quad U = \begin{pmatrix} 1 & -2 & 4 & -8 & 16 \\ 0 & 4 & 0 & 16 & 0 \\ 0 & 0 & -4 & 0 & -16 \\ 0 & 0 & 0 & -3 & -3 \\ 0 & 0 & 0 & 0 & -6 \end{pmatrix} \quad P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Resolviendo los sistemas $Ly = Pb$ y $Ux = y$ obtenemos que $x = \begin{pmatrix} 1 \\ -1/2 \\ 1/2 \\ 1 \\ 0 \end{pmatrix}$. Por lo tanto, tomando estos valores de x como los coeficientes del polinomio interpolante, obtenemos que

$$\boldsymbol{p(x) = x^3 + \frac{1}{2}x^2 - \frac{1}{2}x + 1} \blacksquare$$

c) Para calcular en la forma de diferencias divididas de Newton se empleó una rutina propia para generar todas las diferencias divididas de la siguiente manera.

Dados n puntos para interpolar de la forma $\{(x_i, f(x_i))\}_{i=1}^{n}$. Recordemos que las diferencias divididas se definen como

$$f[x_i, \cdots, x_j] = \frac{f[x_{i+1}, \cdots, x_j] - f[x_i, \cdots, x_{j-1}]}{x_j - x_i}, 1 \le i < j \le n$$
$$f[x_i] = f(x_i), 1 \le i \le n$$

Mientras que los polinomios de Newton se definen como

$$N_m(x) = N_{m-1}(x) \cdot (x - x_{m-1}), 2 \le m \le n$$
$$N_1(x) = 1$$

Entonces el polinomio interpolante en la forma de Newton es

$$p(x) = \sum_{i=1}^{n} f[x_1, \cdots, x_i] N_1(x) = f[x_1]N_1(x) + f[x_1, x_2]N_2(x) + \cdots + f[x_1, \cdots, x_n]N_n(x)$$

Por lo que, podemos hacer uso de un sencillo procedimiento recursivo para determinar las diferencias que son realmente necesarias $f[x_1], \cdots, f[x_1, \cdots, x_n]$ como se muestra:

```
%Hacemos que los puntos interpolados (x,y) y la matriz D sean
globales para usarlos en la función recursiva
global x y D;
%Se introducen en X y en Y los puntos (x_i,y_i) a interpolar
x=[-2;-1;0;1;2]; y=[-4;1;1;2;10]; n=length(x);
%D es la matriz de las diferencias divididas
D=zeros(n); D(:,1)=y;
%Llamamos a la función desde la entrada (n,n) e imprimimos la
diagonal
dife_divi(n,n); coef=diag(D); disp(coef);

%Función recursiva para construir las diferencias divididas
function [res]=dife_divi(i,j)
    global x y D;
    %Si es la primera columna, el caso es trivial f[x_i]=y(i)
    if j==1
        res=y(i);
        return;
    end
    %Sino, debe computarse a partir de los anteriores
    D(i,j)=dife_divi(i,j-1)-dife_divi(i-1,j-1);
    D(i,j)=D(i,j)/(x(i)-x(i-(j-1)));
    res=D(i,j);
    return;
end
```

En este ejemplo en concreto se obtiene

$$f[x_1] = -4,\ \ f[x_1, x_2] = 5,\ \ f[x_1, \cdots, x_3] = -\frac{5}{2},\ \ f[x_1, \cdots, x_4] = 1,\ \ f[x_1, \cdots, x_5] = 0$$

De modo que el polinomio interpolante en la forma de Newton es

$$\begin{aligned} p(x) &= -4(1) + 5(x+2) - \frac{5}{2}(x+2)(x+1) + 1(x+2)(x+1)(x) \\ &\quad + 0(x+2)(x+1)(x)(x-1) \\ \Longrightarrow p(x) &= -4 + 5x + 10 - \frac{5}{2}x^2 - \frac{15}{2}x - \frac{10}{2} + x^3 + 3x^2 + 2x \\ \therefore \boldsymbol{p(x)} &= \boldsymbol{x^3 + \frac{1}{2}x^2 - \frac{1}{2}x + 1} \end{aligned}$$

3.17 Computa la tabla de diferencias divididas y $p_3(x)$ para los datos

x	0	$\pi/2$	π
f	0	1	0

Verifica que este polinomio es el mismo que en la forma de Lagrange.

Para realizar este ejercicio sencillamente hacemos una pequeña modificación de la rutina recursiva mostrada en el ejercicio anterior. Concretamente cambiamos los puntos a interpolar y esta vez imprimimos toda la matriz D de las diferencias divididas.

```
Command Window
>> diferencias_divididas
         0            0            0
         1      0.63662            0
         0     -0.63662     -0.40528
```

Analíticamente podemos comprobar que

$$f[x_1] = 0,\quad f[x_1, x_2] = \left(\frac{1-0}{\frac{\pi}{2}-0}\right) = \frac{2}{\pi},\quad f[x_1, x_2, x_3] = \left(\frac{-\frac{2}{\pi}-\frac{2}{\pi}}{\pi - 0}\right) = -\frac{4}{\pi^2}$$

Entonces el polinomio interpolante en la forma de Newton es

$$p(x) = 0(1) + \frac{2}{\pi}(x) - \frac{4}{\pi^2}(x)\left(x - \frac{\pi}{2}\right) = \frac{2}{\pi}x - \frac{4}{\pi^2}x^2 + \frac{2}{\pi}x$$

$$\therefore \boldsymbol{p(x) = -\frac{4}{\pi^2}x^2 + \frac{4}{\pi}x = -\frac{4}{\pi^2}x(x-\pi)}$$

Que efectivamente, resulta ser igual que el polinomio interpolante en la forma de Lagrange para este mismo problema.