EPFL

Numerical Analysis and Computational Mathematics

Fall Semester 2019 - Section CSE

Dr. Rafael Vázquez Hernández

Assistant: Ondine Chanon

Session 5 - 16 October 2019

Solutions - Approximation of functions and data

Solution I (MATLAB, tutorial)
Solutions are reported in the tutorial.

Solution II (MATLAB)

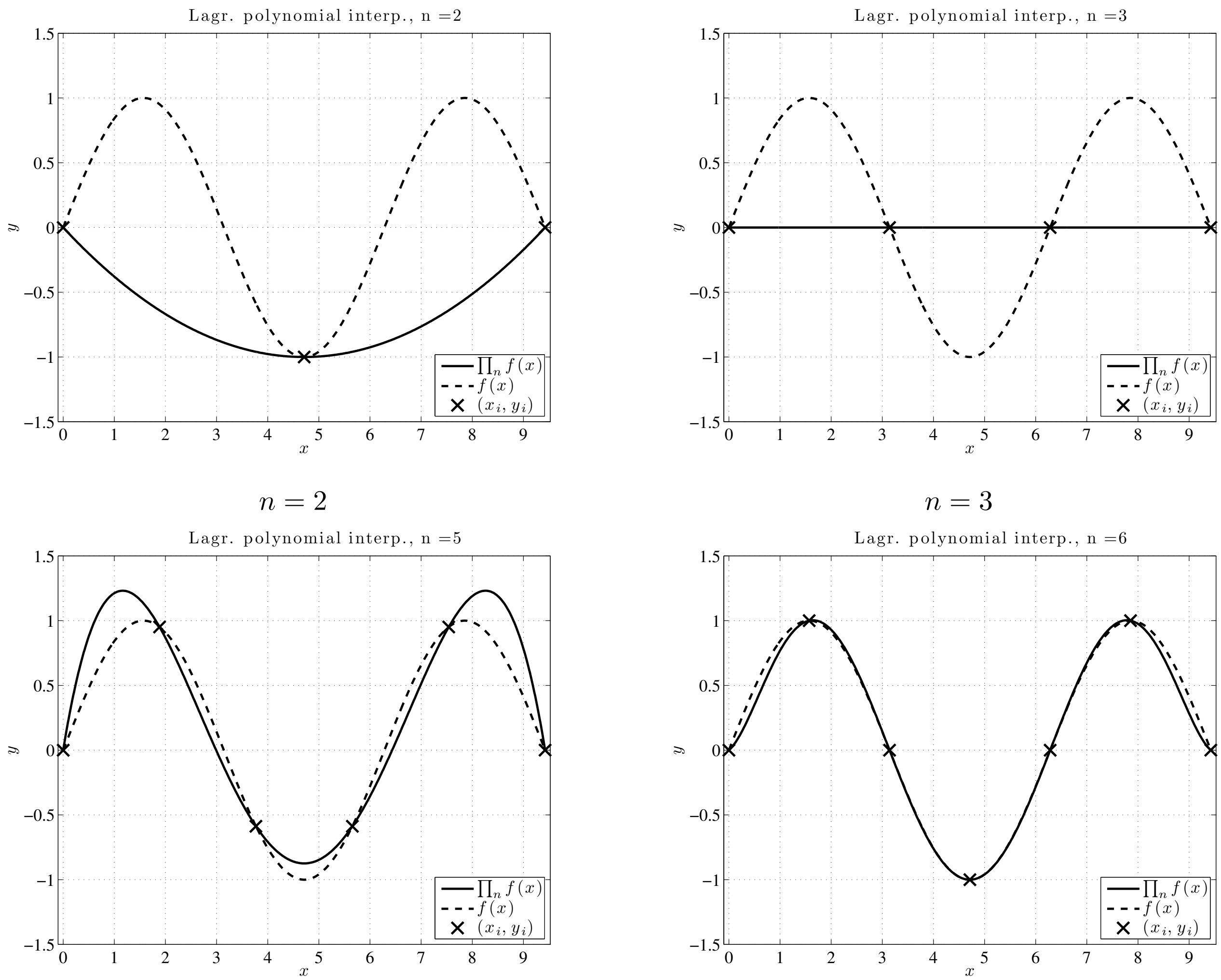
a) We execute the following commands:

```
f = @(x) sin( x );   a = 0;   b = 3 * pi;
n_vect = 1 : 7;   % vector containing all the degrees of desired polynomials
x_values = linspace( a, b, 1001 );
f_values = f( x_values );
for n = n_vect    % for all the degrees in n_vect
    x_nodes = linspace( a, b, n + 1 );
    y_nodes = f( x_nodes );
    P = polyfit( x_nodes, y_nodes, n );
    P_values = polyval( P, x_values );
    figure( n );
    plot( x_values, P_values, '-k', ...
              x_values, f_values, '--k', x_nodes, y_nodes, 'xk'  );
    legend( '\Pi_n f(x)', 'f(x)', '(x_i,y_i)');
end
```

We obtain the results reported in Figure 1 for example for $n = 2, 3, 5$, and 6. We observe the convergence of the interpolating polynomials $\Pi_n f(x)$ to $f(x)$ for increasing values of n. For $n = 3$ we observe that the couples $\{(x_i, f(x_i))\}$ for $i = 0, 1, 2, 3$ are aligned on a straight, horizontal line for which we obtain $\Pi_3 f(x) = c \in \mathbb{R}$, i.e. a polynomial of degree $n = 0$; specifically, we obtain that $\Pi_3 f(x) = 0$.

b) We insert the computation of the error in the MATLAB commands executed at point a), e.g.:

```
f = @(x) sin( x );   a = 0;   b = 3 * pi;
n_vect = 1 : 7;   % vector containing all the degrees of desired polynomials
x_values = linspace( a, b, 1001 );
```

$n = 2$ $n = 3$

$n = 5$ $n = 6$

Figure 1: Interpolating polynomials $\Pi_n f(x)$ of the function $f(x) = \sin(x)$ at equally spaced nodes in $I = [0, 3\pi]$ for $n = 2, 3, 5$, and 6.

```
f_values = f( x_values );
err = [ ];   % initialization of the vector containing the true errors
for n = n_vect
    x_nodes = linspace( a, b, n + 1 );
    y_nodes = f( x_nodes );
    P = polyfit( x_nodes, y_nodes, n );
    P_values = polyval( P, x_values );
    err = [ err, max( abs( P_values - f_values ) ) ];  % append errors to err
end
err
%  err =
%      1.0000    1.5925    1.0000    0.6363    0.4228    0.1301    0.0895
plot( n_vect, err, '-ko' );
```

As we can observe from Figure 1(left) the error $e_n(f)$ is decreasing when n is increasing.

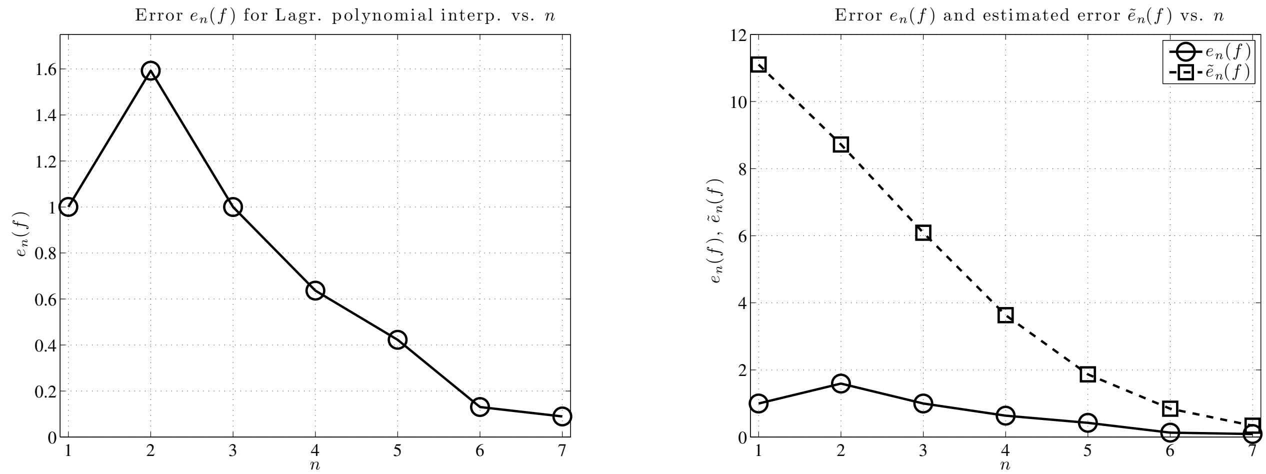


Figure 2: Errors $e_n(f)$ vs. n for the interpolating polynomials $\Pi_n f(x)$ of the function $f(x) = \sin(x)$ (left) and comparison with the error estimators $\widetilde{e}_n(f)$ (right).

c) We observe that $\max_{x\in I} \left|f^{(n+1)}(x)\right| = 1$, since $f^{(1)}(x) = \cos(x)$, $f^{(2)}(x) = -\sin(x)$, $f^{(3)}(x) = -\cos(x)$,..., and $x \in I = [0, 3\pi]$. As consequence, the error estimator reads $\widetilde{e}_n(f) = \frac{1}{4(n+1)}\left(\frac{b-a}{n}\right)^{n+1}$, which is monotonically decreasing when n increases. We plot in Figure 2(right) the error estimator $\widetilde{e}_n(f)$ in comparison with the error $e_n(f)$ by means of the following commands:

```
err_estimated = [ ];
for n = n_vect
       df_max = 1;  % for all n and x \in I=[0,3 *pi]
       err_estimated = [ err_estimated, ...
             1 / ( 4 * ( n + 1 ) ) * ( ( b - a ) / n )^( n + 1 ) * df_max ];
end
err_estimated
%  err_estimated =
%     11.1033    8.7205    6.0881    3.6310    1.8689    0.8427    0.3375
plot( n_vect, err, '-ko', n_vect, err_estimated, '--ks' );
```

We verify that $e_n(f) \leq \widetilde{e}_n(f)$ for all n. Since $\lim_{n\to\infty} \widetilde{e}_n(f) = 0$ we have that $\lim_{n\to\infty} e_n(f) = 0$, i.e. the polynomial $\Pi_n f(x)$ converges to $f(x)$ for all $x \in I$ by increasing n.

Solution III (MATLAB)

a) The Lagrange interpolating polynomial of degree n for $f(x)$ is $\Pi_n f(x) = \sum_{k=0}^{n} f(x_k)\,\varphi_k(x)$, where $\varphi_k(x) := \prod_{i=0,\, i\neq k}^{n} \frac{x-x_i}{x_k-x_i}$ are the Lagrange characteristic functions and x_i distinct nodes with $i = 0, \ldots, n$. For $n = 2$, we calculate $\varphi_k(x)$ for $k = 0, 1, 2$ as:

$$\varphi_0(x) = \frac{x - x_1}{x_0 - x_1}\frac{x - x_2}{x_0 - x_2} = x^2 - \frac{5}{2}x + 1,$$
$$\varphi_1(x) = \frac{x - x_0}{x_1 - x_0}\frac{x - x_2}{x_1 - x_2} = -\frac{4}{3}x^2 + \frac{8}{3}x,$$
$$\varphi_2(x) = \frac{x - x_0}{x_2 - x_0}\frac{x - x_1}{x_2 - x_1} = \frac{1}{3}x^2 - \frac{1}{6}x.$$

By observing that $f(x_0) = -2$, $f(x_1) = -\frac{11}{8}$, and $f(x_2) = 2$, we obtain $\Pi_2 f(x) = \frac{1}{2}x^2 + x - 2$.

b) In this case we have $\varphi_0(x) = \frac{1}{2}x^2 - \frac{3}{2}x + 1$, $\varphi_1(x) = -x^2 + 2x$, and $\varphi_2(x) = \frac{1}{2}x^2 - \frac{1}{2}x$. By observing that $f(x_0) = -2$, $f(x_1) = 0$, and $f(x_2) = 2$, we obtain $\Pi_2 f(x) = 2x - 2$ which is a polynomial of degree 1. The result is due to the fact that the couples $\{(x_i, f(x_i))\}$ for $i = 0, \ldots, n$ are aligned on a straight line.

c) It is sufficient to observe that $f(x)$ is polynomial of degree 3 and therefore we have $\Pi_3 f(x) \equiv f(x)$.

Solution IV (MATLAB)

a) We execute the following commands to compare the interpolating polynomials $\Pi_n f(x)$ with $f(x)$ in Figure 3.

```
f = @(x) 1 ./ ( 1 + x.^2 );    a = -5;    b = 5;
n_vect = [ 2 4 8 12 ];
x_values = linspace( a, b, 1001 );
f_values = f( x_values );
for n = n_vect
   x_nodes = linspace( a, b, n + 1 );
   y_nodes = f( x_nodes );
   P = polyfit( x_nodes, y_nodes, n );
   P_values = polyval( P, x_values );
   figure( n );
   plot( x_values, P_values, '-k', ...
           x_values, f_values, '--k', x_nodes, y_nodes, 'xk' );
   legend( '\Pi_n f(x)', 'f(x)', '(x_i,y_i)');
end
```

We observe that oscillations of the polynomials $\Pi_n f(x)$ appear at the extrema of the interval I for n "large", thus highlighting the so called Runge phenomenon; the amplitude of these oscillations generally increases with n.

b) We plot the error $e_n(f)$ vs. n in Figure 4 by modifying the MATLAB commands at point a):

```
err = [ ];
for n = n_vect
   x_nodes = linspace( a, b, n + 1 );
   y_nodes = f( x_nodes );
   P = polyfit( x_nodes, y_nodes, n );
   P_values = polyval( P, x_values );
```

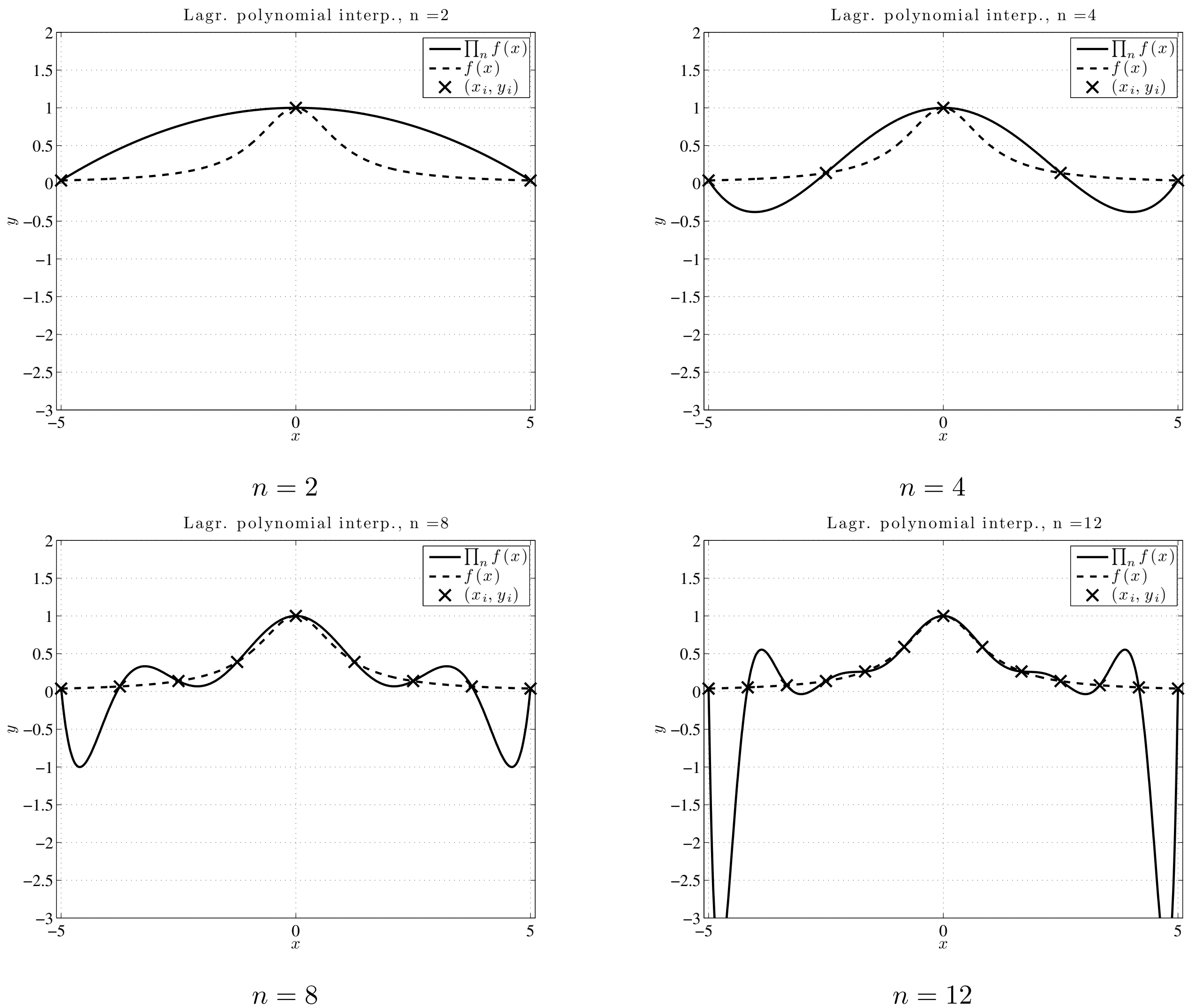


Figure 3: Interpolating polynomials $\Pi_n f(x)$ of the function $f(x) = \frac{1}{1+x^2}$ at equally spaced nodes in $I = [-5, 5]$ for $n = 2, 4, 8$, and 12.

```
   err = [ err, max( abs( P_values - f_values ) ) ];
end
err
%  err =
%      0.6462    0.4384    1.0452    3.6630
figure; plot( n_vect, err, '-ko' );
```

Accordingly to the comments at point a), we see that the error $e_n(f)$ increases for n increasing due to the insurgency of the Runge phenomenon. The latter is a phenomenon which may occur when the value of $\max_{x\in I} \left|f^{(n+1)}(x)\right|$ considerably increases with n.

c) We repeat point a) by using the Chebyshev–Gauss–Lobatto nodes in place of the equally spaced ones to determine the corresponding interpolating polynomials, which we denote as $\Pi_n^c f(x)$. In MATLAB, we use the following commands to obtain the results of Figure 5.

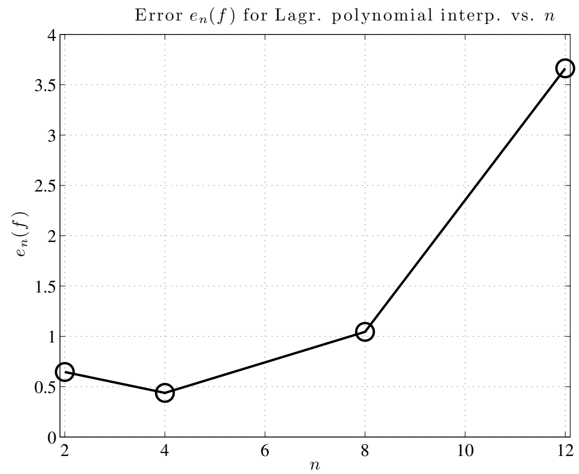


Figure 4: Errors $e_n(f)$ vs. n for interpolating polynomials $\Pi_n f(x)$ of the function $f(x) = \frac{1}{1+x^2}$ at equally spaced nodes in $I = [-5, 5]$; $n = 2, 4, 8$, and 12.

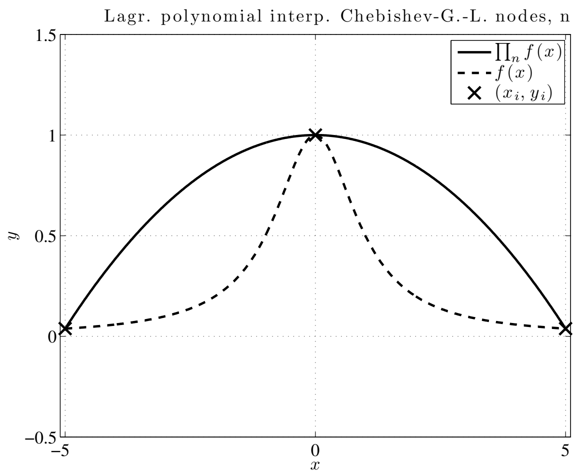
```
for n = n_vect
   x_nodes_c = (a+b)/2 + (b-a)/2 * ( - cos( pi * [ 0 : n ] / n ) );
   y_nodes_c = f( x_nodes_c );
   P_c = polyfit( x_nodes_c, y_nodes_c, n );
   P_c_values = polyval( P_c, x_values );
   figure( n + 100);
   plot( x_values, P_c_values, '-k', ...
         x_values, f_values, '--k', x_nodes_c, y_nodes_c, 'xk' );
   legend( '$\prod_n f(x)$', '$f(x)$', '$(x_i,y_i)$' );
end
```

We observe that the interpolating polynomials $\Pi_n^c f(x)$ converge to $f(x)$ for increasing values of n. In Figure 6 we compare the interpolating polynomials $\Pi_8^c f(x)$ and $\Pi_8 f(x)$ with $f(x)$.

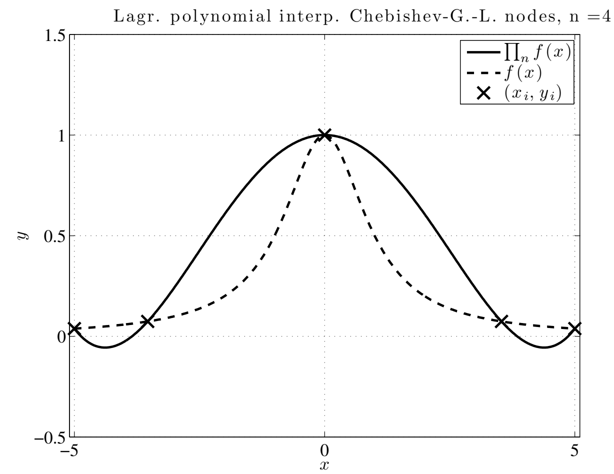
d) By repeating point b) for the Chebyshev–Gauss–Lobatto nodes, we obtain that the error $e_n^c(f)$ associated to $\Pi_n^c f(x)$ decreases for increasing values of n (see Figure 7). We use the following MATLAB commands.

```
err_c = [ ];
for n = n_vect
   x_nodes_c = (a+b)/2 + (b-a)/2 * ( - cos( pi * [ 0 : n ] / n ) );
   y_nodes_c = f( x_nodes_c );
   P_c = polyfit( x_nodes_c, y_nodes_c, n );
   P_values_c = polyval( P_c, x_values );
   err_c = [ err_c, max( abs( P_values_c - f_values ) ) ];
end
err_c
%  err_c =
%     6.4623e-01   4.5998e-01   2.0468e-01   8.4396e-02
plot( n_vect, err_c, '-ks'  );
```

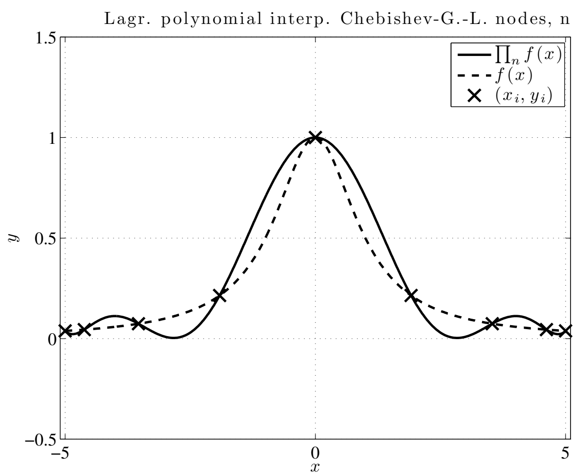
The result is justified by the fact that the use of the Chebyshev–Gauss–Lobatto nodes ensures that $\lim_{n\to\infty} e_n^c(f) = 0$ for $f(x) \in C^{(n+1)}(I)$.



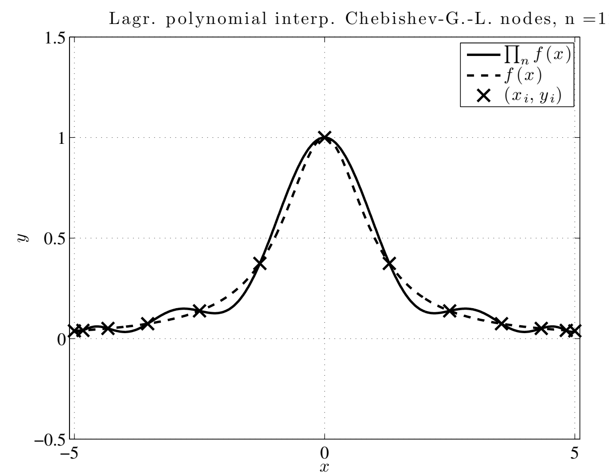
$n = 2$



$n = 4$



$n = 8$



$n = 12$

Figure 5: Interpolating polynomials $\Pi_n f(x)$ of the function $f(x) = \frac{1}{1+x^2}$ at the Chebyshev–Gauss–Lobatto nodes in $I = [-5, 5]$ for $n = 2, 4, 8$, and 12.

Solution V (Theoretical)

a) In general, given a function $f(x) \in C^{(n+1)}(I)$ with $I = [a, b]$ and the corresponding interpolating polynomial $\Pi_n f(x)$ of degree n defined at $n+1$ equally spaced nodes $\{x_i\}_{i=0}^n$, we have the following estimate for the error $e_n(f) := \max_{x\in I} |f(x) - \Pi_n f(x)|$:

$$e_n(f) \leq \widetilde{e}_n(f), \qquad \text{where } \widetilde{e}_n(f) := \frac{1}{4(n+1)} \left(\frac{b-a}{n}\right)^{n+1} \max_{x\in I} \left|f^{(n+1)}(x)\right|.$$

Specifically, for $f(x) = \sin\left(\frac{x}{3}\right)$, we obtain that $f^{(1)}(x) = \frac{1}{3}\cos\left(\frac{x}{3}\right)$, $f^{(2)}(x) = -\frac{1}{9}\sin\left(\frac{x}{3}\right)$, $f^{(3)}(x) = -\frac{1}{27}\cos\left(\frac{x}{3}\right)$, ...; as consequence, since $I = [a, b] = [0, 1]$, we deduce that $\max_{x\in I} \left|f^{(n+1)}(x)\right| \leq \frac{1}{3^{n+1}}$. By renaming $\widetilde{e}_n(f)$ and using the previous result, we obtain that:

$$e_n(f) \leq \widetilde{e}_n(f), \qquad \text{with } \widetilde{e}_n(f) = \frac{1}{4(n+1)(3n)^{n+1}}.$$

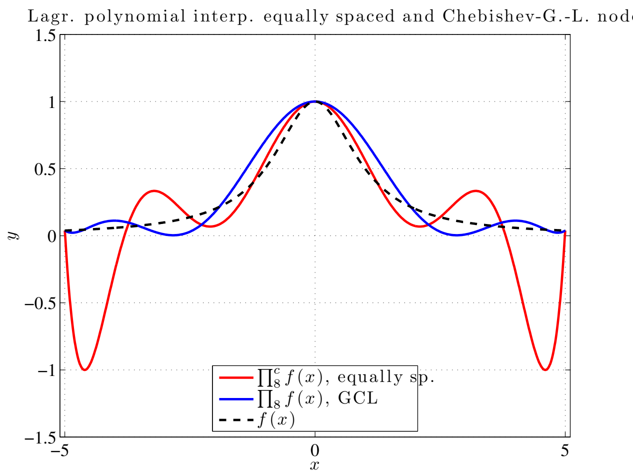


Figure 6: Interpolating polynomials $\Pi_8^c f(x)$ and $\Pi_8 f(x)$ of the function $f(x) = \frac{1}{1+x^2}$ at the Chebyshev–Gauss–Lobatto and equally spaced nodes in $I = [-5, 5]$, respectively.

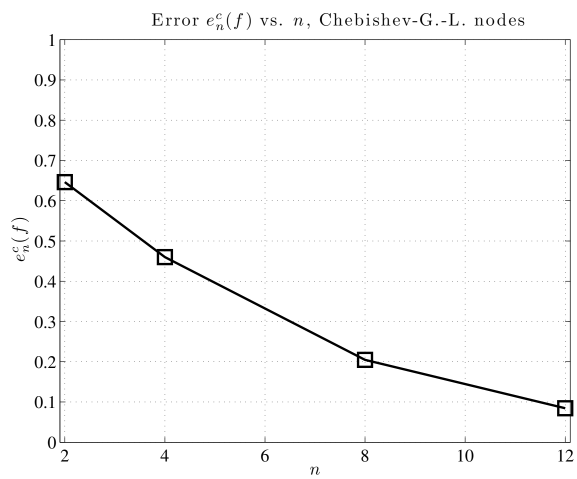


Figure 7: Errors $e_n^c(f)$ vs. n for interpolating polynomials $\Pi_n^c f(x)$ of the function $f(x) = \frac{1}{1+x^2}$ at the Chebyshev–Gauss–Lobatto nodes in $I = [-5, 5]$; $n = 2, 4, 8$, and 12.

Since $\lim_{n\to\infty} \widetilde{e}_n(f) = 0$, we have that the error $e_n(f)$ tends to zero for increasing values of n, i.e. $\lim_{n\to\infty} e_n(f) = 0$.

b) We proceed by trial and error evaluating $\widetilde{e}_n(f)$ for $n = 1, 2, 3, \ldots$ until we guarantee that $e_n(f) \leq \widetilde{e}_n(f) < 10^{-4}$. We obtain $\widetilde{e}_1(f) = 1.3889 \cdot 10^{-2}$, $\widetilde{e}_2(f) = 3.8580 \cdot 10^{-4}$, and, finally, $\widetilde{e}_3(f) = 9.5260 \cdot 10^{-6}$. As consequence, the minimum number of equally spaced nodes in I necessary to ensure that $e_n(f) < 10^{-4}$ is $n + 1 = 4$.

c) The Chebyshev–Gauss–Lobatto nodes in $I = [a, b]$ are determined by the formula:

$$x_i = \frac{a+b}{2} + \frac{b-a}{2}\,\widehat{x}_i, \qquad \text{where } \widehat{x}_i := -\cos\left(\frac{\pi}{n} i\right), \qquad \text{for } i = 0, \ldots, n.$$

For $n = 3$, we have $\widehat{x}_0 = -1$, $\widehat{x}_1 = -\frac{1}{2}$, $\widehat{x}_2 = \frac{1}{2}$, $\widehat{x}_2 = 1$. Since $a = 0$ and $b = 1$, we obtain $x_0 = 0$, $x_1 = \frac{1}{4}$, $x_2 = \frac{3}{4}$, $x_2 = 1$.

d) Since the Chebyshev–Gauss–Lobatto nodes are not equally spaced in I, we consider the following error estimate for the interpolating polynomials $\Pi_n f(x)$:

$$e_n(f) \leq \widetilde{e}_n(f), \qquad \text{where } \widetilde{e}_n(f) := \frac{1}{(n+1)!} \max_{x \in I} \left| f^{(n+1)}(x) \right| \max_{x \in I} |\omega_n(x)|.$$

By setting $n = 3$, selecting $f(x) = \sin\left(\frac{x}{3}\right)$ in $I = [0, 1]$, observing that $\max_{x \in I} |\omega_3(x)| < 0.016$ (from Figure 1 of the exercises) and $\max_{x \in I} \left| f^{(4)}(x) \right| \leq \frac{1}{3^4}$ (from point a)), and renaming $\widetilde{e}_3(f)$, we obtain that $e_3(f) \leq \widetilde{e}_3(f)$, with $\widetilde{e}_3(f) = 8.2305 \cdot 10^{-6}$. As consequence, the error $e_3(f)$ associated to the interpolating polynomial $\Pi_3 f(x)$ of the function $f(x)$ at the Chebyshev–Gauss–Lobatto nodes is inferior or equal to $8.2305 \cdot 10^{-6}$.