Polynomial approximation of functions and derivatives

The vast majority of numerical methods for ODEs and PDEs rely on polynomial approximation of derivatives and integrals. In this course note we review well-known (and also not so well-known) results in polynomial approximation theory of smooth functions. To this end, denote by

$$\mathbb{P}_n([a,b]) = \text{span}\{1, x, \dots, x^n\} \tag{1}$$

the space of polynomial of degree at most n defined on the interval $[a,b]$. It is well-known that any continuous function $f(x)$ defined on $[a,b]$ can be approximated by a polynomial $p_n(x) \in \mathbb{P}_n([a,b])$ as close as we like, where "close" is in the sense of the uniform norm[1]. This result is summarized in the following theorem.

Theorem 1 (Weierstrass (1885))**.** Let $f \in C_0([a,b])$ (continuous function defined on an interval $[a,b]$). Then for any $\epsilon > 0$ there exists $n_\epsilon \in \mathbb{N}$ and a polynomial $p_{n_\epsilon}(x) \in \mathbb{P}_n([a,b])$ such that

$$\|f - p_{n_\epsilon}\|_\infty = \max_{x\in[a,b]} |f(x) - p_{n_\epsilon}(x)| \leq \epsilon. \tag{3}$$

This theorem guarantees the existence of an appropriate sequence of polynomials converging uniformly to any given continuous function , but it does not provide a constructive way to determine $p_{n_\epsilon}(x)$. There are many different methods one can use to approximate a function in terms of polynomials, e.g., *interpolation methods*, *projection methods*, *least squares*, etc. In this course note we discuss primarily interpolation methods.

Polynomial interpolation

Consider a continuous function $f(x)$ defined on an interval $[a,b]$. Choose $n+1$ *distinct* points $\{x_0, \dots, x_n\}$ in $[a,b]$ (interpolation nodes) and let $y_i = f(x_i)$ be the value of $f(x)$ at x_i. The following theorem guarantees the existence of a unique polynomial of degree less or equal than n that interpolates $f(x)$ at the nodes $\{x_0, \dots, x_n\}$.

Theorem 2. Let $f \in C_0([a,b])$ and consider $n+1$ distinct nodes $\{x_0, \dots, x_n\}$ in $[a,b]$. Then, there exists a unique polynomial $\Pi_n f(x)$ of degree less or equal than n that interpolates f at $\{x_0, \dots, x_n\}$, i.e.,

$$\Pi_n f(x_i) = f(x_i) \qquad i = 0, \dots, n. \tag{4}$$

Proof. Consider the $n+1$ pairs $\{x_i, y_i\}$ where $y_i = f(x_i)$ $(i = 0, \dots, n)$ and the polynomial

$$\Pi_n f(x) = a_0 + a_1 x + \dots a_n x^n \qquad a_i \in \mathbb{R}. \tag{5}$$

By imposing the interpolation conditions $\Pi_n f(x_i) = y_i$ we obtain the following linear system of $(n+1)$ equations in $(n+1)$ unknowns $\{a_0, \dots, a_n\}$

$$\underbrace{\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix}}_{\text{Vandermonde matrix } V} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}. \tag{6}$$

[1] Let $f(x)$ be a continuous function in $[a,b]$. The uniform (or L^∞) norm of f is defined as

$$\|f\|_\infty = \max_{x\in[a,b]} |f(x)|. \tag{2}$$

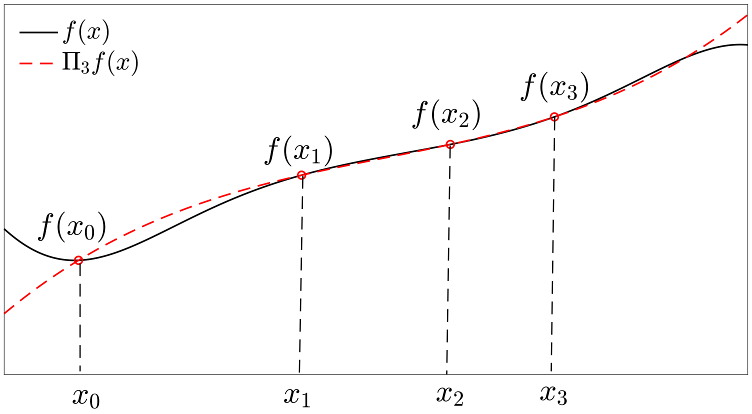

Figure 1: Sketch of a function $f(x)$ and the polynomial $\Pi_3 f(x)$ of degree 3 that interpolates $f(x)$ at $\{x_0, \ldots, x_3\}$.

The determinant of the Vandermonde matrix V can be expressed as

$$\det(V) = \prod_{0 \leq i < j \leq n} (x_i - x_j) \, . \tag{7}$$

Hence, if $x_i \neq x_j$ for $i \neq j$ (distinct nodes) then $\det(V) \neq 0$. This implies that the system (6) has a unique solution, i.e., that there exist a unique polynomial of degree at most n that interpolates $f(x)$ at $\{x_0, \ldots, x_n\}$.

□

The Vandermonde matrix is usually ill-conditioned, with few notable exceptions, e.g., trigonometic interpolation problems on evenly-spaced grids. Therefore, computing the interpolating polynomial via solution of the linear system (6) is often not advisable.

Lagrangian interpolation

Lagrangian interpolation relies on representing the polynomial interpolant $\Pi_n f(x)$ in terms of a polynomial basis defined *uniquely* by the set of grid points $\{x_0, \ldots, x_n\}$. Such basis has the form

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \qquad \text{(Lagrange characteristic polynomial).} \tag{8}$$

By evaluating $l_i(x)$ at $\{x_0, \ldots, x_n\}$ we see that, by construction, $l_i(x)$ is a polynomial of degree n that interpolates the following dataset

$$\boldsymbol{x} = \{x_0, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n\} \qquad\qquad \boldsymbol{y} = \{0, \ldots, 0, \underset{\substack{\uparrow \\ i\text{-th entry}}}{1}, 0, \ldots, 0\}. \tag{9}$$

In other words,

$$l_i(x_j) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{10}$$

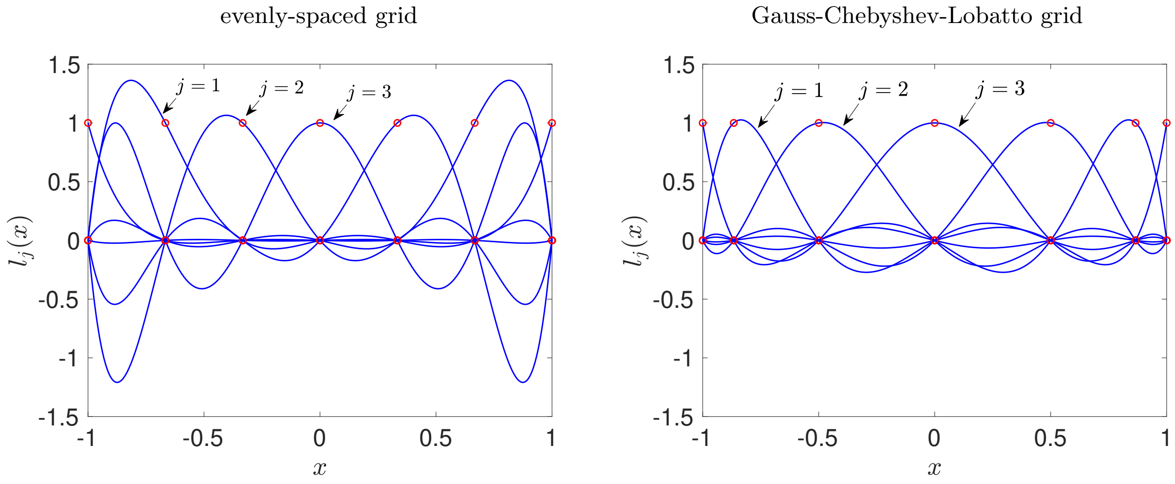

Figure 2: Lagrange characteristic polynomials corresponding to evenly-spaced and Gauss-Chebyshev-Lobatto (GCL) grids with 7 points in $[-1, 1]$.

It is straightforward to show that $\{l_0(x), \ldots, l_n(x)\}$ are *linearly independent* and therefore $\{l_0(x), \ldots, l_n(x)\}$ is a cardinal[2] basis for $\mathbb{P}_n$ (space of polynomials of degree at most n).

With the Lagrange characteristic polynomials available, it is very easy to compute the (unique) polynomial that interpolates $f(x)$ at the nodes $\{x_0, \ldots, x_n\}$. In fact, since $\{l_0(x), \ldots, l_n(x)\}$ is a basis for $\mathbb{P}_n$, we have that the linear combination representing $\Pi_n f(x)$ is unique. Moreover, since $\{l_0(x), \ldots, l_n(x)\}$ is a cardinal basis we have that the coefficients of the linear combination are simply the values of f at the grid points $\{x_0, \ldots, x_n\}$, i.e.,

$$\Pi_n f(x) = \sum_{k=0}^{n} f(x_k) l_k(x) \qquad \text{(Lagrange interpolation formula).} \tag{11}$$

Let us verify that $\Pi_n f(x)$ indeed interpolates $f(x)$ on the grid $\{x_0, \ldots, x_n\}$. To this end, we simply evaluate $\Pi_n f$ at an arbitrary node x_j to obtain

$$\Pi_n f(x_j) = \sum_{k=0}^{n} f(x_k) l_k(x_j) = \sum_{k=0}^{n} f(x_k)\delta_{kj} = f(x_j), \tag{12}$$

i.e., $\Pi_n f$ interpolates f at x_j $(j = 0, \ldots, n)$.

Remark: The Lagrange characteristic polynomial $l_j(x)$ can be also written in terms of the so-called *nodal polynomial*

$$\omega_{n+1}(x) = \prod_{k=0}^{n} (x - x_k) \tag{13}$$

as

$$l_j(x) = \frac{\omega_{n+1}(x)}{(x - x_j)\omega'_{n+1}(x_j)}. \tag{14}$$

Remark: The Lagrange characteristic polynomials (8) (or (14)) depend exclusively on the interpolation nodes $\{x_0, \ldots, x_n\}$ and they can be computed numerically by solving the polynomial interpolation problem

[2] A basis defined by some set of grid points is called *cardinal* if each basis element is equal to one at one grid points and zero at all other points.

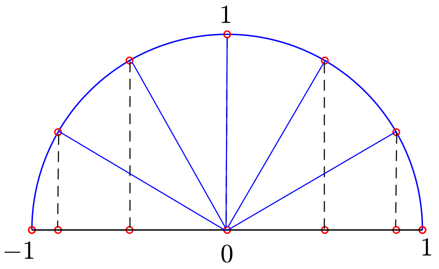

Figure 3: Geometric construction of Gauss-Chebyshev-Lobatto points.

(6) for the dataset (9). To minimize the condition number of the Vandermonde matrix V it is advisable to perform a preliminary scaling of the interpolation nodes. In Figure 2 we plot the Lagrange characteristic polynomials corresponding to an evenly-spaced grid with 7 points in $[-1, 1]$, and a Gauss-Chebyshev-Lobatto grid with the same number of points

$$x_j = -\cos\left(\frac{j\pi}{6}\right) \qquad j = 0, \dots, 6. \tag{15}$$

The GCL points are obtained by dividing a half circle with radius 1 centered at the origin into equally spaced sectors (in this case 6) and projecting the corresponding points onto the x-axis. In Appendix A we show that CGL points are zeros of the polynomial

$$Q_{n+1}(x) = (1 - x^2)\frac{dT_n(x)}{dx}, \tag{16}$$

where $T_n(x)$ is n-th degree Chebyshev polynomial of the first kind. In Figure 3 we set $n = 6$.

Interpolation error

The interpolation error incurred when replacing a function $f(x)$ by its polynomial interpolant $\Pi_n f$ depends on the function and on the location (and number) of the interpolation nodes $\{x_0, \dots, x_n\}$. The following Theorem provides an exact analytical expression for the pointwise interpolation error (see [2, p. 335]).

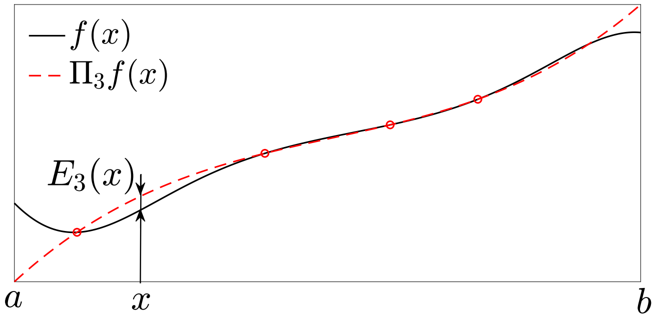

Figure 4: Pointwise interpolation error when approximating $f(x)$ with a third-order polynomial in $[a, b]$.

Theorem 3. Let $f \in C^{n+1}([a, b])$ and $\Pi_n f(x)$ the polynomial of degree n interpolating $f(x)$ at the $n + 1$ distinct nodes $\{x_0, \dots, x_n\}$ in $[a, b]$. Then for any $x \in I[a, b]$

$$E_n(x) = f(x) - \Pi_n f(x) = \frac{f^{(n+1)}(\xi)}{(n + 1)!}\omega_{n+1}(x), \tag{17}$$

where $f^{(n+1)}(\xi)$ is the $(n+1)$-th derivative of f evaluated at some point $\xi \in [a,b]$ and

$$\omega_{n+1}(x) = \prod_{k=0}^{n}(x - x_k) \qquad \text{(nodal polynomial).} \tag{18}$$

Theorem 3 provides the exact interpolation error in terms of the $(n+1)$-th derivative of the function $f(x)$ evaluated at some point $\xi \in [a,b]$ and the nodal polynomial (18). Taking the uniform norm[3] of (17) yields

$$\|f(x) - \Pi_n f(x)\|_\infty = \frac{\left|f^{(n+1)}(\xi)\right|}{(n+1)!}\|\omega_{n+1}(x)\|_\infty . \tag{20}$$

Evenly-spaced grids. Consider an evenly-spaced interpolation grid in $[a,b]$ with $n+1$ points

$$x_j = a + \frac{j}{n}(b-a) \qquad j = 0, \dots, n. \tag{21}$$

The nodal polynomial at the right hand side of (20) can be bounded as

$$\begin{aligned}\|\omega_{n+1}(x)\|_\infty &= \max_{x\in[a,b]}\left|\prod_{k=0}^{n}(x - x_j)\right| \\ &= \max_{x\in[a,b]}\prod_{k=0}^{n}|(x - x_j)| \\ &\leq n!\Delta x^{n+1},\end{aligned} \tag{22}$$

where $\Delta x = (b-a)/n$ is the uniform grid spacing. The last inequality can be proved by noting that the maximum of the product of the absolute values $|(x - x_j)|$ can be bounded when x is close to one of the endpoints of the interval $[a,b]$. Substituting (22) into (20) yields

$$\|f - \Pi_n f\|_\infty \leq \Delta x^{n+1}\frac{\left|f^{(n+1)}(\xi)\right|}{(n+1)} \leq \Delta x^{n+1}\frac{\left\|f^{(n+1)}\right\|_\infty}{(n+1)}. \tag{23}$$

Hence, convergence of polynomial interpolation on evenly-spaced grids is granted if Δx^{n+1} goes to zero faster (as we increase n) than $\left\|f^{(n+1)}\right\|_\infty/(n+1)$. Note that for a *fixed* number of evenly-spaced nodes, i.e., fixed n convergence is granted if we send $(b-a)$ to zero. In fact,

$$\|f - \Pi_n f\|_\infty \leq C(b-a)^{n+1}\left\|f^{(n+1)}\right\|_\infty \qquad \text{where} \qquad C = \frac{1}{n^{n+1}(n+1)}, \tag{24}$$

which goes to zero for fixed n if $(b-a) \to 0$. This happens, for example, if we interpolate a function $f(x)$ *locally*, e.g., via piecewise polynomial interpolation within a small spatial domain. Clearly, if $\left\|f^{(n+1)}\right\|_\infty$ is *bounded* as $n \to \infty$ then Lagrangian interpolation on evenly-spaced grids converges in the uniform norm as $n \to \infty$.

[3] Recall that the uniform norm of a continuous function f on an interval $[a,b]$ is defined as

$$\|f\|_\infty = \max_{x\in[a,b]}|f(x)| . \tag{19}$$

Lebesgue constant. The effects of grid spacing on the accuracy of polynomial interpolation can be also studied via the following general theorem.

Theorem 4. Let $f \in C^0([a,b])$ and $\Pi_n f(z)$ the polynomial of degree n interpolating $f(x)$ at the $n+1$ distinct nodes $\{x_0, \ldots, x_n\}$. Then

$$\|f(x) - \Pi_n f(x)\|_\infty \leq (1 + \Lambda_n) \inf_{\Psi_n \in \mathbb{P}_n} \|f(x) - \Psi_n(x)\|_\infty \tag{25}$$

where

$$\Lambda_n = \max_{x\in[a,b]} \lambda_n(x) \qquad \text{(Lebesgue constant)}, \tag{26}$$

$$\lambda_n(z) = \sum_{j=0}^{n} |l_j(x)| \qquad \text{(Lebesgue function)}. \tag{27}$$

Proof. Let $\Psi \in \mathbb{P}_n$ be the best approximating polynomial

$$\|f(x) - \Pi_n f(x)\|_\infty \leq \|f(x) - \Psi_n(x)\|_\infty + \|\Psi_n(x) - \Pi_n f(x)\|_\infty . \tag{28}$$

At this point, we represent $\Psi_n(z)$ and $\Pi_n f(z)$ in terms of the same set of Lagrange polynomials associated with the grid $\{x_0, \ldots, x_n\}$ to obtain

$$\begin{aligned}\|\Psi_n(x) - \Pi_n f(z)\|_\infty &= \left\| \sum_{j=0}^{n} \left[\Psi(x_j) - f(x_j)\right] l_j(x) \right\|_\infty \\ &\leq \|\Psi_n(x) - f(x)\|_\infty \underbrace{\max_{x\in[a,b]} \sum_{j=0}^{n} |l_j(x)|}_{\Lambda_n} . \end{aligned} \tag{29}$$

A substitution of (29) into (28) yields (25).

□

Estimates on the Lebesgue constant

The Lebesgue constant depends exclusively on the interpolation nodes $\{x_0, \ldots, x_n\}$ (see equations (26)-(27) and equation (8)). Clearly, the smaller the Lebesgue constant, the smaller the upper bound on the polynomial interpolation error (25).

It can be shown that, no matter how we choose the points, the Lebesgue constant grows at least logarithmically with n, i.e.,

$$\Lambda_n \geq \frac{2}{\pi} \log(1+n) + C \quad \text{as} \quad n \to \infty, \tag{30}$$

where C is a constant independent of n (see [1, p. 102]). This does not mean that the interpolation error necessarily grows with n. It just means that the upper bound in (25) diverges as $n \to \infty$, i.e., that we cannot grant uniform convergence of Lagrangian interpolation based on (25). For any given sequence of grid points one can find continuous functions for which the polynomial interpolant will exhibit non-uniform convergence. On the other hand, one can also show that for any given continuous function one can always construct a set of grid points that will result in a uniformly convergent polynomial approximations. Thus, we cannot (in general) seek one set of grid points x_j that will exhibit optimal behavior for all possible interpolation problems. However, the behavior of the Lebesgue constant can serve as a guideline to understand whether certain families of grid points are likely to result in well-behaved interpolation polynomials.

It is possible to bound the Lebesgue constant corresponding to different types of grids. For instance, for evenly-spaced grids of $n+1$ points in $[-1,1]$, i.e.,

$$x_j = -1 + \frac{2j}{n} \qquad j = 0,\ldots,n \tag{31}$$

we have

$$\frac{2^{n-2}}{n^2} \leq \Lambda_n \leq \frac{2^{n+3}}{n}. \tag{32}$$

Similarly, for the Gauss-Chebyshev-Lobatto (GCL) grid (see Appendix A and Figure 3)

$$x_j = -\cos\left(\frac{j\pi}{n}\right) \qquad j = 0,\ldots,n \tag{33}$$

we have

$$\Lambda_n \leq \frac{2}{\pi}\log(n) + B \qquad \text{(finite } n\text{)}, \tag{34}$$

where B is a suitable constant independent of n [1, p. 105].

Example: In Figure 5 we plot the Lagrangian interpolant of the function

$$f(x) = \frac{1}{1+10x^2} \qquad x \in [-1,1] \tag{35}$$

computed at 17 evenly-spaced nodes, 17 Gauss-Chebyshev-Lobatto (GCL) nodes, and 17 Gauss-Chebyshev-Lobatto nodes in $[-1,1]$. We also plot the Lebesgue functions corresponding to all three interpolation grids. The Lebesgue constants for the evenly-spaced, GCL, and GLL grids, i.e., the maxima of the functions $\lambda_n(x)$ displayed in Figure 5 are, respectively

$$\Lambda_n^{\text{even}} = 934.53 \qquad \Lambda_n^{\text{GCL}} = 2.72, \qquad \Lambda_n^{\text{GLL}} = 2.47. \tag{36}$$

If we measure the interpolation error corresponding to GLL grid in the $L^2([-1,1])$ norm instead of the uniform norm we used in Theorem 4, then it is possible to obtain a *spectral convergence* result. For example, the following convergence result holds for Gauss-Legendre and Gauss-Legendre-Lobatto interpolation (see Table 3 in Appendix A and [1, p. 114]).

Theorem 5. Let $f(x) \in H^s([-1,1])$, $s \geq 1$. Then

$$\|f - \Pi_n f\|_{L^2([-1,1])} \leq C n^{-s} \|f(x)\|_{H^s([-1,1])} \tag{37}$$

where $\{x_0,\ldots,x_n\}$ are either Gauss-Legendre points or Gauss-Legendre-Lobatto points (see Table 3).

In Theorem 5, $H^s([-1,1])$ denotes the Sobolev space of degree s, which is defined to be the space of functions with square integrable (weak) derivatives up to order s and norm

$$\|f(x)\|^2_{H^s([-1,1])} = \int_{-1}^{1}\left[f^2(x) + \sum_{k=1}^{s}\left(\frac{df(x)}{dx}\right)^2\right]dx < \infty. \tag{38}$$

Note that the $L^2([-1,1])$ interpolation error for Gauss-Legendre and Gauss-Legendre-Lobatto grids depends on the *degree of smoothness* of the function f we are approximating, i.e., the exponent s in (37). This type of convergence is called *spectral convergence*. If f is of class C^∞ (i.e. infinitely smooth, $s \to \infty$) then convergence can be *exponential*, i.e., the interpolation error (37) goes to zero, e.g., as $e^{-\alpha n}$, for some positive $\alpha > 0$. In Figure 6 we demonstrate such exponential converge for the GLL interpolant $\Pi_n f$ of the function f defined in (35) as the number of GLL points $(n+1)$ increases.

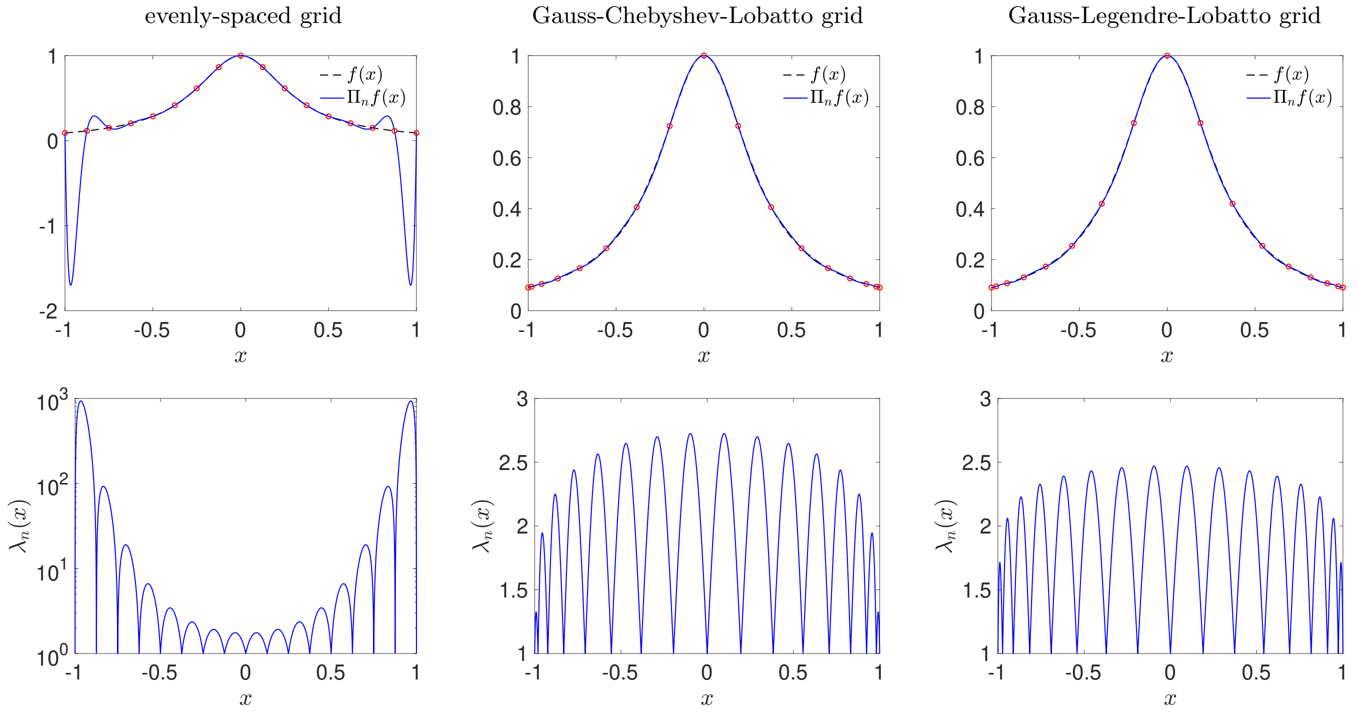

Figure 5: Lagrangian interpolation of $f(x) = (1+10x^2)^{-1}$ using 17 evenly-spaced nodes (left), and 17 Gauss-Chebyshev-Lobatto (GCL) nodes (center), and 17 Gauss-Legendre-Lobatto (GLL) nodes. The Lebesgue functions $\lambda_n(x)$ associated with the evenly-spaced, GCL and GLL grids have maxima $\Lambda_n^{\text{even}} = 934.53$ and $\Lambda_n^{\text{GCL}} = 2.72$, $\Lambda_n^{\text{GLL}} = 2.47$, respectively.

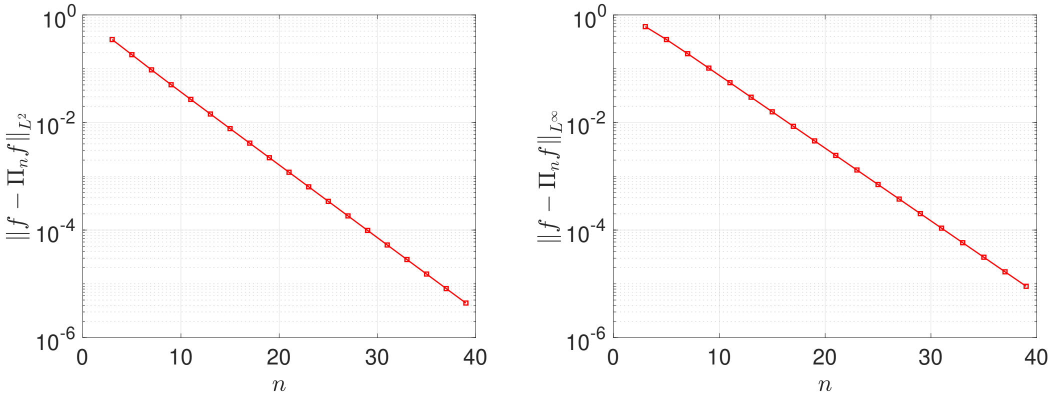

Figure 6: Exponential converge for the GLL interpolant of the function (35) as n increases. Shown are results in the L^2 norm (left) and L^∞ (i.e., uniform) norm (right).

Piecewise polynomial interpolation

Rather than looking for a polynomial interpolant that approximates $f(x)$ globally within an interval $[a, b]$, we can look for an interpolant made of local polynomial patches. The union of such patches can be a continuous (piecewise differentiable) function as in the case of piecewise linear interpolation, or a smooth function of class $C^{(k-1)}([a, b])$ as in the case of *interpolatory splines* [2, p. 355] (see Figure 7).

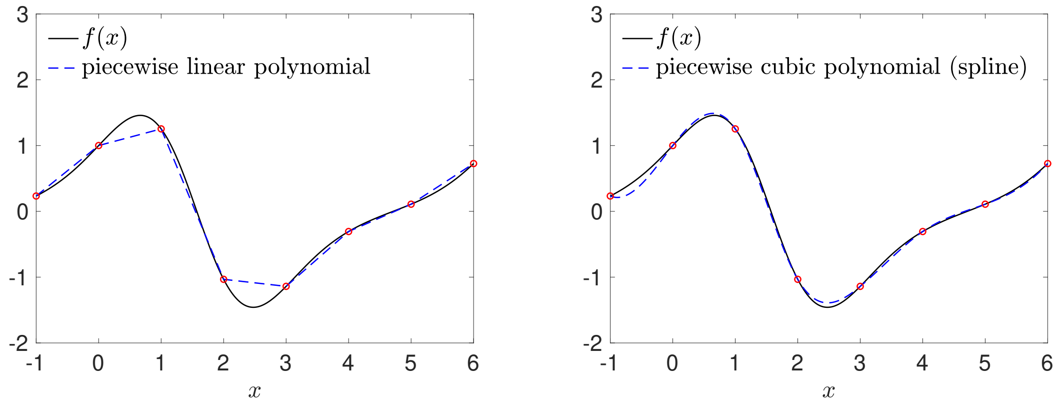

Figure 7: Piecewise polynomial interpolation with linear and cubic polynomials (interpolatory cubic spline). The interpolation nodes are shown with red circles.

Numerical differentiation

To compute the derivative of a function $f(x)$ numerically, we first approximate f using simpler functions, e.g., local or global interpolating polynomials. Subsequently, we differentiate these polynomials and utilize their derivatives as an approximation of the derivative of the function f. In this section, we discuss two numerical differentiation methods, i.e.,

- finite differences,
- pseudo-spectral methods.

Both methods are based on polynomial interpolation. In particular, finite difference methods are based on local interpolation (usually at evenly-spaced grids), while pseudo-spectral methods are based on polynomial interpolants at Gauss or Gauss-Lobatto points.

Finite differences

Finite differences are numerical differentiation methods based on replacing $f(x)$ *locally* with a Lagrangian interpolating polynomial $\Pi_n f(x) \simeq f(x)$, which is then used to approximate the derivative of $f(x)$ as

$$\frac{df(x)}{dx} \simeq \frac{d\Pi_n f(x)}{dx}, \tag{39}$$

for all x in the neighborhood of a desired point x_k. The local interpolating polynomial $\Pi_n f(x)$ is usually constructed on a evenly-spaced grid defined in a neighborhood of x_k. Of course, x_k can also be (and it usually is) a grid point.

First-order finite differences. The simplest finite-difference formulas are based on piecewise linear interpolation of f in a neighborhood of a point x_k. With reference to Figure 8(a), the derivative of f at the grid point x_k can be approximated by two different first-order polynomials, i.e., the line interpolating the points $\{(x_k, f(x_k)), (x_{k+1}, f(x_{k+1})\}$ (line 2), and the line interpolating $\{(x_{k-1}, f(x_{k-1})), (x_k, f(x_k)\}$ (line 1). This gives us two different approximations of the derivative df/dx at x_k, namely

$$\frac{df(x_k)}{dx} \simeq \frac{f(x_k+\Delta x) - f(x_k)}{\Delta x} \qquad \text{(forward)}, \tag{40}$$

$$\frac{df(x_k)}{dx} \simeq \frac{f(x_k) - f(x_k-\Delta x)}{\Delta x} \qquad \text{(backward)}. \tag{41}$$

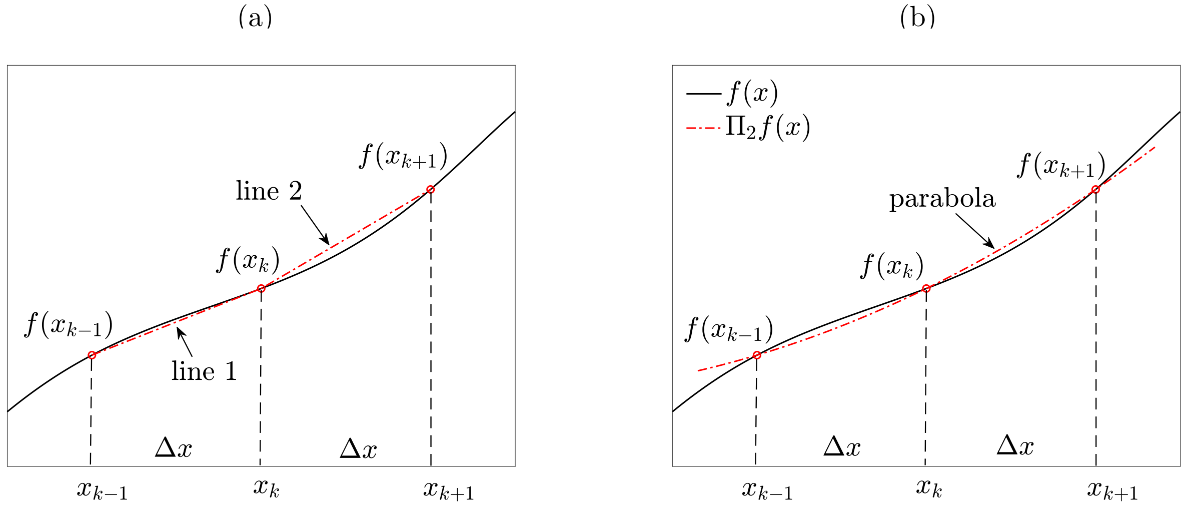

Figure 8: Interpolating polynomials for first-order (a) and second-order (b) finite-difference approximations of the derivatives of $f(x)$.

The attributes "forward" and "backward" refer to the location of the points we are using to approximate the derivative at x_j. In the case of forward finite-difference formula we are using the function value at $x_{k+1} = x_k + \Delta x$ which is "to the right" (hence "forward") relative to x_k.

Clearly, both forward and backward finite difference formulas converge to the derivative $f'(x_k)$ as $\Delta x \to 0$. At which rate in Δx? Assuming that $f \in C^2([a, b])$ we can expand the numerator at the right hand side of (40)-(41) in a Taylor series to obtain

$$f(x_k + \Delta x) - f(x_k) = f'(x_k)\Delta x + \frac{1}{2} f''(\xi_1)(\Delta x)^2, \tag{42}$$

$$f(x_k) - f(x_k - \Delta x) = f'(x_k)\Delta x - \frac{1}{2} f''(\xi_1)(\Delta x)^2, \tag{43}$$

where $\xi_1 \in [x_k, x_k + \Delta x]$ and $\xi_2 \in [x_k - \Delta x, x_k]$. Substituting these expressions back into (40)-(41) yields

$$\left| \frac{df(x_k)}{dx} - \frac{f(x_k + \Delta x) - f(x_k)}{\Delta x} \right| = \frac{1}{2} \left| f''(\xi_1) \right| \Delta x, \tag{44}$$

$$\left| \frac{df(x_k)}{dx} - \frac{f(x_k) - f(x_k - \Delta x)}{\Delta x} \right| = \frac{1}{2} \left| f''(\xi_2) \right| \Delta x. \tag{45}$$

Hence, both finite-difference approximations (40)-(41) are *first-order* in Δx, meaning that the error in the approximation goes to zero linearly in Δx as we send Δx to zero. Higher-order finite-difference formulas are obtained similarly by approximating $f(x)$ locally with an interpolating polynomial of higher degree.

Second-order finite differences. We approximate $f(x)$ locally using an interpolating polynomial of degree two. With reference to Figure 8(b) we have the Lagrangian interpolation formula[4]

$$\begin{aligned} \Pi_2 f(x) =& f(x_{j-1}) l_{j-1}(x) + f(x_j) l_j(x) + f(x_{j+1}) l_{j+1}(x) \\ =& f(x_{j-1}) \frac{(x - x_j)(x - x_{j+1})}{2\Delta x^2} - f(x_j) \frac{(x - x_j)(x - x_{j+1})}{\Delta x^2} + f(x_{j+1}) \frac{(x - x_{j-1})(x - x_j)}{2\Delta x^2}. \end{aligned} \tag{47}$$

[4] We have, for example

$$l_{j-1}(x) = \frac{(x - x_j)}{\underbrace{(x_{j-1} - x_j)}_{-\Delta x}} \frac{(x - x_{j+1})}{\underbrace{(x_{j-1} - x_{j+1})}_{-2\Delta x}} = \frac{(x - x_j)(x - x_{j+1})}{2\Delta x^2}. \tag{46}$$

At this point we differentiate $\Pi_2 f(x)$ to obtain

$$\frac{d\Pi_2 f(x)}{dx} =\frac{f(x_{j-1})}{2\Delta x^2}\left(2x - x_j + x_{j+1}\right) - \frac{f(x_j)}{\Delta x^2}\left(2x - x_{j-1} + x_{j+1}\right) + \frac{f(x_{j+1})}{2\Delta x^2}\left(2x - x_{j-1} + x_j\right). \tag{48}$$

Depending on where we evaluate $d\Pi_2 f(x)/dx$, we have have different differentiation formulas. Specifically,

$$f'(x_j) \simeq \frac{d\Pi_2 f(x_j)}{dx} =\frac{f(x_{j+1}) - f(x_{j-1})}{2\Delta x} \qquad \text{(centered)}, \tag{49}$$

$$f'(x_{j_1}) \simeq \frac{d\Pi_2 f(x_{j-1})}{dx} =\frac{-3f(x_{j-1}) + 4f(x_j) - f(x_{j+1})}{2\Delta x} \qquad \text{(forward)}, \tag{50}$$

$$f'(x_{j+1}) \simeq \frac{d\Pi_2 f(x_{j+1})}{dx} =\frac{3f(x_{j+1}) - 4f(x_j) + f(x_{j-1})}{2\Delta x} \qquad \text{(backward)}. \tag{51}$$

Let us now prove that the centered finite difference formula (51) is indeed second-order accurate in Δx as $\Delta x \to 0$. To this end, we need to show that

$$\left| f'(x_j) - \frac{f(x_{j+1}) - f(x_{j-1})}{2\Delta x} \right| = O\left((\Delta x)^2\right). \tag{52}$$

To this end, we expand the finite difference formula in (52) to third-order in Δx about x_j. We have

$$f(x_{j+1}) =f(x_j) + f'(x_j)\Delta x + \frac{f''(x_j)}{2}(\Delta x)^2 + \frac{f'''(\xi_1)}{6}(\Delta x)^3, \tag{53}$$

$$f(x_{j-1}) =f(x_j) - f'(x_j)\Delta x + \frac{f''(x_j)}{2}(\Delta x)^2 - \frac{f'''(\xi_2)}{6}(\Delta x)^3, \tag{54}$$

where $\xi_1 \in [x_j, x_{k+1}]$ and $\xi_2 \in [x_{j-1}, x_j]$. Subtracting (54) from (53) yields

$$\begin{aligned}\frac{f(x_{j+1}) - f(x_{j-1})}{2\Delta x} =&\frac{1}{2\Delta x}\left[2f'(x_j)\Delta x + \frac{(\Delta x)^3}{6}\left(f'''(\xi_1) - f'''(\xi_2)\right)\right]\\ =&f'(x_j) + \frac{(\Delta x)^2}{12}\left(f'''(\xi_1) + f'''(\xi_2)\right).\end{aligned} \tag{55}$$

Hence,

$$\left| f'(x_j) - \frac{f(x_{j+1}) - f(x_{j-1})}{2\Delta x} \right| = \frac{(\Delta x)^2}{12}\left|f'''(\xi_1) + f'''(\xi_2)\right|, \tag{56}$$

which goes to zero quadratically in Δx. Note also that as we send Δx to 0, both ξ_1 and ξ_2 converge to x_j (from above and below, respectively). Next, we compute an approximation of the second derivative of f. This is achieved by differentiating $d\Pi_2 f(x)/dx$ in (48) once more with respect to x. This yields

$$\frac{d^2\Pi_2 f(x)}{dx^2} =\frac{f(x_{j-1}) - 2f(x_j) + f(x_{j+1})}{\Delta x^2} \qquad \text{for all } x \in [x_{j-1}, x_{j+1}]. \tag{57}$$

Using Taylor series it is straightforward to show that (57) approximates $f''(x_j)$ to second-order in Δx, i.e.,

$$\left| f''(x_j) - \frac{f(x_{j-1}) - 2f(x_j) + f(x_{j+1})}{\Delta x^2} \right| = O((\Delta x)^2). \tag{58}$$

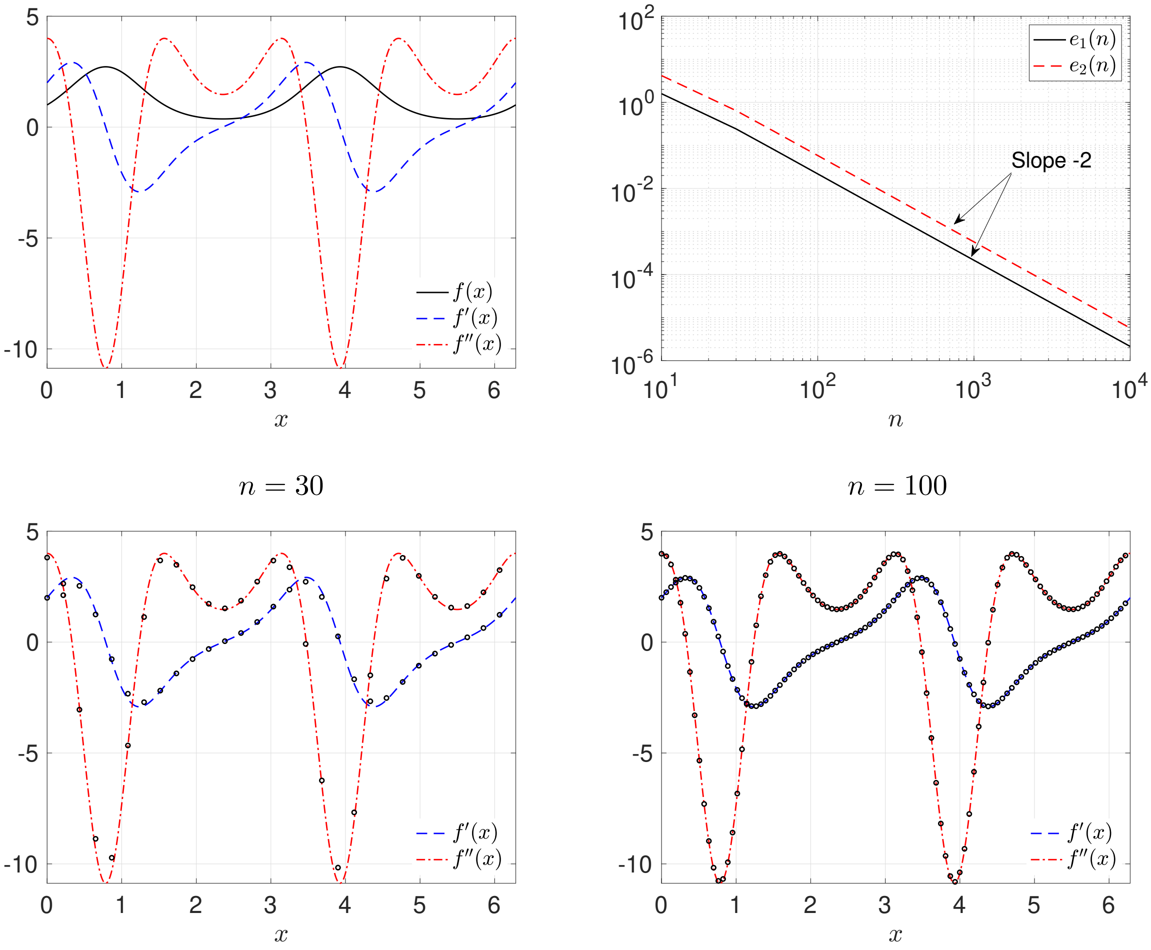

Figure 9: Approximation of the first and the second derivative of the $f(x)$ defined in (63) using second-order centered finite differences (equations (49) and (57)). In (b) we plot the errors (65)-(66) in a log-log scale. It is seen that the errors decay quadratically with n, i.e., they go to zero quadratically as $\Delta x = 2\pi/n$ goes to zero. We also plot the finite-difference approximation of the derivatives $f'(x)$ and $f''(x)$ we obtain for $n = 30$ and $n = 100$ (black circles).

In fact, by summing up the Taylor series

$$f(x_{j+1}) = f(x_j) + f'(x_j)\Delta x + \frac{f''(x_j)}{2}(\Delta x)^2 + \frac{f'''(x_j)}{6}(\Delta x)^3 + \frac{f''''(\xi_1)}{24}(\Delta x)^4, \tag{59}$$

$$f(x_{j-1}) = f(x_j) - f'(x_j)\Delta x + \frac{f''(x_j)}{2}(\Delta x)^2 - \frac{f'''(x_j)}{6}(\Delta x)^3 + \frac{f''''(\xi_2)}{24}(\Delta x)^4, \tag{60}$$

where $\xi_1 \in [x_j, x_{k+1}]$ and $\xi_2 \in [x_{j-1}, x_j]$, we obtain

$$f(x_{j+1}) + f(x_{j-1}) - 2f(x_j) = f''(x_j)(\Delta x)^2 + \frac{f''''(\xi_1)}{24}(\Delta x)^4. \tag{61}$$

This implies that

$$\left| f''(x_j) - \frac{f(x_{j-1}) - 2f(x_j) + f(x_{j+1})}{\Delta x^2} \right| = \frac{(\Delta x)^2}{24} \left| f''''(\xi_1) + f''''(\xi_2) \right|, \tag{62}$$

which goes to zero quadratically in Δx as $\Delta x \to 0$.

Example: Let us compute the second-order finite-difference approximation of the first and second derivative of the periodic function

$$f(x) = e^{\sin(2x)} \qquad x \in [0, 2\pi]. \tag{63}$$

To this end, we construct an evenly-spaced grid in $[0, 2\pi]$ and use the centered finite-difference formulas (49) and (57) with periodic conditions

$$f(x_{n+1}) = f(x_1), \qquad f(x_{-1}) = f(x_n). \tag{64}$$

In Figure 9 we plot the errors

$$e_1(n) = \max_{j=0,\dots,n} \left| f'(x_j) - \frac{f(x_{j+1}) - f(x_{j-1})}{2\Delta x} \right|, \tag{65}$$

$$e_2(n) = \max_{j=0,\dots,n} \left| f''(x_j) - \frac{f(x_{j-1}) - 2f(x_j) + f(x_{j+1})}{\Delta x^2} \right|. \tag{66}$$

relative to the exact derivatives

$$f'(x_j) = 2\cos(2x_j)e^{\sin(2x_j)}, \qquad f''(x_j) = 4\left[\cos^2(2x_j) - \sin(2x_j)\right] e^{\sin(2x_j)}. \tag{67}$$

versus n (number of points in $[0, 2\pi]$ in a log-log scale. Clearly, the errors (65)-(66) decay with n as $1/n^2$. To see this, consider the sequence

$$e(n) = \frac{\kappa}{n^2}, \tag{68}$$

which obviously goes to zero as n^{-2}. Taking the logarithm of (68) yields

$$\log(e) = \log(\kappa) - 2\log(n). \tag{69}$$

Hence if the plot of $\log(e)$ versus $\log(n)$ is line with slope -2 then $e(n)$ decays as $1/n^2$.

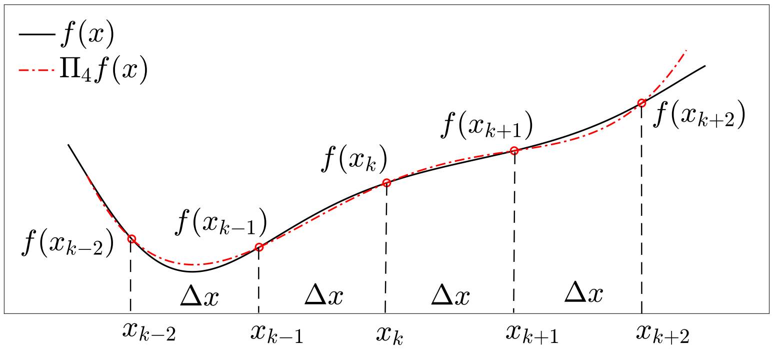

Figure 10: Five-point stencil for the fourth-order finite difference approximation of the derivatives of $f(x)$.

Higher-order finite differences.

By using finite-difference stencils with an increasing number of points it is possible to derive higher order finite-difference formulas. In Table 1 we summarize the coefficients for centered and forward finite-difference approximations of derivatives up to order four. The coefficients of the backward finite difference formulas for odd derivatives are the opposite of the forward ones, while the coefficients for even derivatives are the same (see Table 2). As an example, the fourth-order centered, forward, and backward finite difference approximations of the second derivative of f at x_j are

$$f''(x_j) \simeq \frac{-f(x_{j-2}) - 12f(x_{j-1}) - 30f(x_j) + 12f(x_{j+1}) - f(x_{j+2})}{12\Delta x^2} \qquad \text{(centered)} \tag{70}$$

$$f''(x_j) \simeq \frac{45f(x_{j+1}) - 154f(x_{j+2}) + 214f(x_{j+3}) - 156f(x_{j+4}) + 61f(x_{j+5}) - 10f(x_{j+6})}{12\Delta x^2} \qquad \text{(forward),} \tag{71}$$

$$f''(x_j) \simeq \frac{45f(x_{j-1}) - 154f(x_{j-2}) + 214f(x_{j-3}) - 156f(x_{j-4}) + 61f(x_{j-5}) - 10f(x_{j-6})}{12\Delta x^2} \qquad \text{(backward).} \tag{72}$$

These formulas are obtained by first interpolating the function $f(x)$ locally with a fourth order polynomial (see Figure 10) and then by approximating the derivatives of f by the derivatives of $\Pi_4 f(x)$ (evaluated at various points).

Pseudo-spectral methods

In this section we briefly describe how to compute numerical derivatives using pseudo-spectral methods. The main idea is the same as finite-differences, i.e., approximate the derivative of $f(x)$ by the derivative of a polynomial interpolant. Differently from finite differences where the polynomial interpolant is constructed locally (usually at evenly-spaced grid points), pseudo-spectral methods are based on global interpolating polynomials at Gauss or Gauss-Lobatto nodes $\{x_0, \ldots, x_n\}$. Such polynomial has the Lagrangian form

$$\Pi_n f(x) = \sum_{j=0}^{n} f(x_j) l_j(x). \tag{73}$$

Differentiating $\Pi_n f(x)$ with respect to x yields

$$\frac{d\Pi_n f(x)}{dx} = \sum_{j=0}^{n} f(x_j) \frac{l_j(x)}{dx}. \tag{74}$$

We approximate the derivative of f at x_p ($p = 0, \ldots, n$) by the derivative of $\Pi_n f(x)$ at x_p, i.e.,

$$f'(x_p) \simeq \sum_{j=0}^{n} D^{(1)}_{pj} f(x_j). \tag{75}$$

The matrix with entries

$$D^{(1)}_{ij} = \frac{dl_j(x_p)}{dx} \tag{76}$$

is called first-order pseudo-spectral differentiation matrix. In Appendix we provide explicit formulas for $D^{(1)}_{ij}$ in the case of Gauss-Chebyshev-Lobatto nodes (equation (106)), and for Gauss-Legendre-Lobatto and Gauss-Legendre nodes (equations (117)-(106)). Regarding the accuracy of pseudo-spectral derivatives, the following general theorem hold for Gauss-Legendre points (see [[1, p. 115]).

Theorem 6. Let $\{x_0, \ldots, x_n\}$ be Gauss-Legendre points in $[-1, 1]$ (see Table 3). For any $f(x) \in H^s([-1, 1])$ with $s \geq 1$ and and every $q \in [0, s]$ there exists a positive constant C independent of n such that

$$\|f - \Pi_n f\|_{H^s([-1,1])} \leq C n^{2q-s+1/2} \|f\|_{H^q([-1,1])}. \tag{77}$$

Centered finite differences

Derivative	Order	-3	-2	-1	0	1	2	3
first derivative	2			-1/2	0	1/2		
	4		1/12	-2/3	0	2/3	-1/12	
	6	-1/60	3/20	3/4	0	3/4	-3/20	1/60
second derivative	2			1	-2	1		
	4		-1/12	4/3	-5/2	4/3	-1/12	
	6	1/90	-3/20	3/2	-49/18	3/2	-3/20	1/90
third derivative	2		-1/2	1	0	1	1/2	
	4	1/8	-1	13/8	0	-13/8	1	-1/8
fourth derivative	2		1	-4	6	-4	1	
	4	-1/6	2	-13/2	28/3	-13/2	2	-1/6

Forward finite differences

Derivative	Order	0	1	2	3	4	5	6	7
first derivative	1	-1	1						
	2	-3/2	2	-1/2					
	3	-11/6	3	-3/2	1/3				
	4	-25/12	4	-3	4/3	-1/4			
second derivative	1	1	-2	1					
	2	2	-5	4	-1				
	4	15/4	-77/6	107/6	-13	61/12	-5/6		
third derivative	1	-1	3	-3	1				
	2	-5/2	9	-12	7	-3/2			
	4	-49/8	29	-461/8	62	-307/8	13	-15/8	
fourth derivative	1	1	-4	-6	-4	1			
	2	3	-14	26	-24	11	-2		
	4	28/3	-111/2	142	-1219/6	176	-185/2	82/3	-7/2

Table 1: Coefficients for centered and forward finite-difference approximations.

Backward finite differences

Derivative	**Order**	**-7**	**-6**	**-5**	**-4**	**-3**	**-2**	**-1**	**0**
first derivative	1							-1	1
	2						-1/2	-2	3/2
	3					-1/3	3/2	-3	11/6
	4				1/4	-4/3	3	-4	25/12
second derivative	1						1	-2	1
	2					-1	4	-5	2
	4			-5/6	61/12	-13	107/6	-77/6	15/4
third derivative	1					-1	3	-3	1
	2				3/2	-7	12	-9	5/2
	4		15/8	-13	307/8	-62	461/8	-29	49/8
fourth derivative	1				1	-4	-6	-4	1
	2			-2	11	-24	26	-14	3
	4	-7/2	82/3	-185/2	176	-1219/6	142	-111/2	28/3

Table 2: Coefficients for backward finite-difference approximation. There is an whole class of ODE solvers known as BDF (backward differentiation formulas) methods that are based on backward finite-differences.
.

The norm at the left and right hand side of (77) is a Sobolev norm defined as

$$\|f\|^2_{H^q([-1,1])} = \int_{-1}^{1} \left(f^2(x) + \sum_{k=1}^{q} \left[\frac{d^k f(x)}{dx^k} \right]^2 \right) dx. \tag{78}$$

Such a norm involves derivatives of f. If we measure the distance between f and $\Pi_n f$ in the Sobolev norm we are actually measuring how well $\Pi_n f$ is approximating f, as well as how well the derivatives of $\Pi_n f(x)$ up to order s are approximating the corresponding derivatives of $f(x)$. Theorem (6) establishes *spectral convergence* of the derivatives of $\Pi_n f$ to the derivatives of $f(x)$ (see the commentary after Theorem 5). A remarkable corollary of Theorem 6 is that if f is of class C^∞ then the error goes to zero exponentially fast with n. This means that the pseudo-spectral method approximates the function and its derivatives with order *infinity* if the function f is of class C^∞ smooth.

Remark: Note that as n increases we are considering interpolating polynomials of higher and higher degree. This is in contrast with finite differences, where we select the polynomial degree that interpolates the function $f(x)$ locally and then we send Δx i.e., $(b-a)/n$ to zero.

Pseudo-spectral differentiation in arbitrary intervals. We have seen in Appendix A that Gauss-Chebyshev and Gauss-Legendre nodes are defined in $[-1, 1]$. How do we differentiate a function f defined in an interval $[a, b]$ using pseudo-spectral methods? To answer this question, consider the linear transformation

$$x = \frac{(b-a)}{2}\eta + \frac{a+b}{2} \qquad \eta \in [-1, 1]. \tag{79}$$

The inverse of (79) is

$$\eta = \frac{2}{(b-a)} \left(x - \frac{a+b}{2} \right) \qquad x \in [a, b]. \tag{80}$$

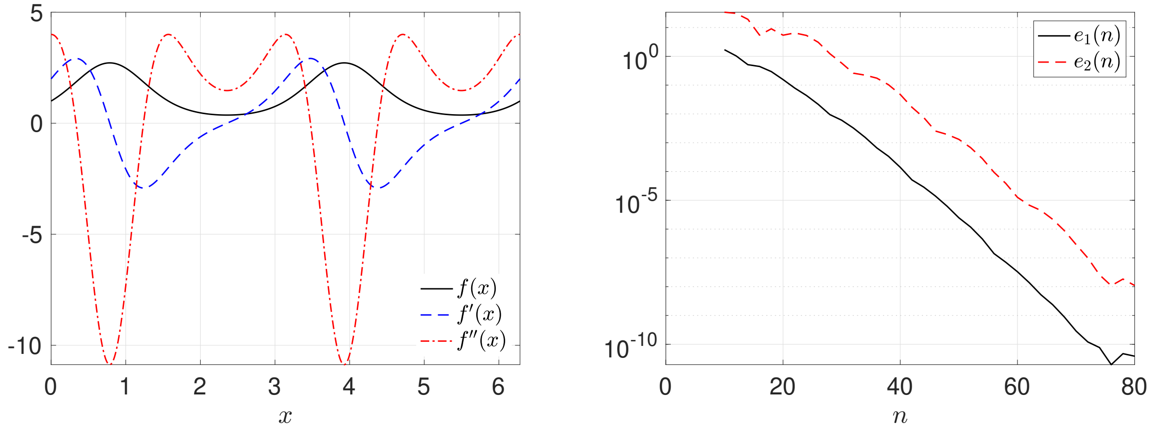

Figure 11: Approximation of the first and the second derivative of the function $f(x)$ defined in (87) using the Gauss-Chebyshev-Lobatto (GCL) pseudo-spectral method. We plot the pointwise errors (91) versus n (number of GCL points). It is seen that the errors decay exponentially fast with the number of points n. Remarkably, with $n = 60$ GCL points we get the same error as with second-order finite differences with about $n = 10000$ points (see Figure 9).

By using (79) we can write $f(x)$ in terms of η as

$$f(x) = f\left(\frac{2}{(b-a)}\left(x - \frac{a+b}{2}\right)\right) \tag{81}$$

This implies that

$$\frac{df}{dx} = \frac{df}{d\eta}\frac{d\eta}{dx} = \frac{2}{(b-a)}\frac{df}{d\eta}. \tag{82}$$

This means that, in practice, if we are interested in differentiating $f(x)$ at Gauss nodes in $[a, b]$ we just need to evaluate $f(x)$ at the nodes

$$x_j = \frac{(b-a)}{2}\eta_j + \frac{a+b}{2} \qquad \eta_j \in [-1, 1]. \tag{83}$$

where $\{\eta_0, \dots, \eta_n\}$ are standard Gauss-Legendre or Gauss-Chebyshev nodes. This yields the column

$$\boldsymbol{f} = \begin{bmatrix} f(x_0) & f(x_1) & \dots & f(x_n). \end{bmatrix}^T \tag{84}$$

A multiplication of $\boldsymbol{f}$ by the re-scaled differentiation matrix

$$\boldsymbol{D}_x^{(1)} = \frac{2}{b-a}\boldsymbol{D}^{(1)}. \tag{85}$$

yields the desired pseudo-spectral derivative at $\{x_0, \dots, x_n\}$. Here, $\boldsymbol{D}^{(1)}$ is the differentiation matrix defined (106) for Gauss-Chebyshev-Lobatto nodes, and in (117)-(106) for Gauss-Legendre-Lobatto and Gauss-Legendre nodes. Similarly, for second-order differentiation matrices we have

$$\frac{d^2 f}{dx^2} = \frac{d^2 f}{d\eta^2}\left(\frac{d\eta}{dx}\right)^2 = \left[\frac{2}{b-a}\right]^2 \frac{d^2 f}{d\eta^2} \quad \Rightarrow \quad \boldsymbol{D}_x^{(2)} = \frac{4}{(b-a)^2}\boldsymbol{D}^{(2)}. \tag{86}$$

Example: To demonstrate the effectiveness of pseudo-spectral differentiation methods, consider again the function (63), i.e.,

$$f(x) = e^{\sin(2x)} \qquad x \in [0, 2\pi], \tag{87}$$

together with its analytical derivatives

$$f'(x) = 2\cos(2x)e^{\sin(2x)}, \qquad f''(x) = 4\left[\cos^2(2x) - \sin(2x)\right]e^{\sin(2x)}. \tag{88}$$

We approximate f' and f'' at Gauss-Chebyshev-Lobatto (GCL) nodes using pseudo-spectral differentiation. To this end, we first construct the vector

$$\boldsymbol{f} = \begin{bmatrix} f(x_0) & f(x_1) & \dots & f(x_n) \end{bmatrix}^T \tag{89}$$

and then apply first- and second-order GCL differentiation matrices $\boldsymbol{D}_x^{(1)}$ and $\boldsymbol{D}_x^{(2)}$ to obtain the approximation of the derivatives at the GCL nodes. The matrices $\boldsymbol{D}_x^{(1)}$ and $\boldsymbol{D}_x^{(2)}$ are rescaled versions (in the sense of (85)-(86)) of the matrices defined in (106) and (108), respectively. In this particular example since $(b-a) = 2\pi$ we have

$$\boldsymbol{D}_x^{(1)} = \frac{1}{\pi}\boldsymbol{D}^{(1)}, \qquad \boldsymbol{D}_x^{(2)} = \frac{1}{\pi^2}\boldsymbol{D}^{(2)}. \tag{90}$$

In Figure 11 we plot the errors

$$e_1(n) = \max_{j=0,\dots,n}\left|f'(x_j) - [\boldsymbol{D}_x^{(1)}\boldsymbol{f}]_j\right|, \qquad e_2(n) = \max_{j=0,\dots,n}\left|f''(x_j) - [\boldsymbol{D}_x^{(2)}\boldsymbol{f}]_j\right| \tag{91}$$

versus the number of GCL points in a semi-logarithmic scale. Clearly, the pseudo-spectral approximation converges *exponentially* fast. This is expected, since the function (87) is infinitely differentiable. Remarkably, with $n = 60$ GCL points we get the same error as with second-order finite differences using $n = 10000$ points (compare Figure 9 and Figure 11)!

Appendix A: Pseudo-spectral interpolation and differentiation

In this appendix we briefly review pseudo-spectral interpolation and differentiation rules based on the Chebyshev and Legendre orthogonal polynomials. These methods are thoroughly described in [1, p. 66-114].

Gauss-Chebyshev Lobatto (GCL) interpolation and differentiation

Consider the trigonometric form of the Chebyshev polynomials of the first kind[5]

$$T_k(x) = \cos(k \arccos(x)) \qquad x \in [-1, 1] \qquad \text{(trigonometric representation).} \tag{95}$$

It can be shown that set $\{T_0(x), T_1(x), \ldots, \}$ satisfies the three-term recurrence relation

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_{k+1}(x) &= 2x\, T_k(x) - T_{k-1}(x). \end{aligned} \tag{96}$$

and the orthogonality conditions

$$\int_{-1}^{1} T_k(x) T_j(x) \underbrace{\frac{1}{\sqrt{1-x^2}}}_{\mu(x)} dx = \delta_{kj} \left\| T_k \right\|^2_{L^2_\mu}. \tag{97}$$

The first few Chebyshev polynomials are given by

$$T_2(x) = 2x^2 - 1, \qquad T_3(x) = 4x^3 - 3x \qquad T_4(x) = 8x^4 - 8x^2 + 1, \ldots. \tag{98}$$

The Gauss-Chebyshev-Lobatto nodes are defined to be the zeros of the polynomial

$$Q_{n+1}(x) = (1 - x^2) \frac{dT_n(x)}{dx}, \tag{99}$$

i.e., they include $x_0 = -1$, $x_n = 1$ and all maxima and minima of the Chebyshev polynomial $T_n(x)$. By differentiating (95) with respect to x we obtain

$$\frac{dT_n(x)}{dx} = \frac{\sin(n \arccos(x))}{\sqrt{1-x^2}}. \tag{100}$$

Hence $Q_{n+1}(x) = 0$ implies that

$$x_j = -\cos\left(\frac{j\pi}{n}\right) \qquad j = 0, \ldots, n \qquad \text{(Gauss-Chebyshev-Lobatto points).} \tag{101}$$

These points are obtained by dividing half unit circle in evenly-spaced parts and projecting them onto the x-axis (see Figure 3). Note also that Chebyshev grid points are nested for $n = 2, 4, 8, \ldots, 2^s$. It can be shown that the Lagrange characteristic polynomials associated with the Gauss-Chebyshev-Lobatto (GCL) nodes are

$$l_j(x) = \frac{(-1)^{n+j+1}(1-x^2)}{d_j n^2 (x - x_j)} \frac{dT_n(x)}{dx} = \frac{(-1)^{n+j+1}\sqrt{(1-x^2)}}{d_j n^2 (x - x_j)} \sin(n \arccos(x)), \tag{102}$$

[5] Note that (95) are indeed polynomials. For example,

$$\cos(\arccos(x)) = x, \tag{92}$$

$$\cos(2\arccos(x)) = 2\left(\cos(\arccos(x))\right)^2 - 1 = 2x^2 - 1, \tag{93}$$

$$\cos(3\arccos(x)) = 4\left(\cos(\arccos(x))\right)^3 - 3\cos(\arccos(x)) = 4x^3 - 3x. \tag{94}$$

where x_j is given in (101) and

$$d_0 = d_n = 2 \qquad d_1 = d_2 = \cdots = d_{n-1} = 1. \tag{103}$$

For any function $f(x)$ defined in $[-1,1]$ we have the following Lagrangian interpolant

$$\Pi_n f(x) = \sum_{k=0}^{n} f(x_k) l_k(x), \qquad x \in [-1,1]. \tag{104}$$

At this point we can differentiate the interpolant (104) and evaluate the derivative at the CGL points (101) to obtain

$$\begin{aligned} \left.\frac{d\Pi_n f(x)}{dx}\right|_{x=x_j} &= \sum_{k=0}^{n} f(x_k) \left.\frac{dl_k(x)}{dx}\right|_{x=x_j} \\ &= \sum_{k=0}^{n} D^{(1)}_{jk} f(x_k), \end{aligned} \tag{105}$$

where $D^{(1)}_{jk}$ is the first-order differentiation matrix

$$D^{(1)}_{ij} = \begin{cases} -\dfrac{2n^2+1}{6} & i=j=0 \\ \dfrac{d_i}{d_j}\dfrac{(-1)^{i+j}}{x_i - x_j} & i \neq j \\ -\dfrac{x_i}{2(1-x_i)^2} & i = j \quad \text{(not 0 or } n\text{)} \\ \dfrac{2n^2+1}{6} & i=j=n \end{cases} \tag{106}$$

and x_i and d_i are defined in (101) and (103), respectively. The expression (105) represents the pseudo-spectral approximation of the derivative $f'(x)$ evaluated at the CGL nodes (101). Similarly, the second derivative of f is approximated by

$$\begin{aligned} \left.\frac{d^2\Pi_n f(x)}{dx^2}\right|_{x=x_j} &= \sum_{k=0}^{n} f(x_k) \left.\frac{d^2 l_k(x)}{dx^2}\right|_{x=x_j} \\ &= \sum_{k=0}^{n} D^{(2)}_{jk} f(x_k), \end{aligned} \tag{107}$$

where $D^{(1)}_{jk}$ is the second-order differentiation matrix

$$D^{(2)}_{ij} = \begin{cases} \dfrac{(-1)^{i+j}}{d_i}\dfrac{x_i^2 + x_i x_j - 2}{\left(1-x_i^2\right)\left(x_i - x_j\right)^2} & 1 \leq i \leq n-1, \quad 0 \leq j \leq n, \qquad j \neq i \\ -\dfrac{(n^2-1)(1-x_i^2)+3}{3\left(1-x_i^2\right)^2} & 1 \leq i = j \leq n-1 \\ \dfrac{2(-1)^j}{3d_j}\left[\dfrac{(2n^2+1)(1-x_j)-6}{\left(1-x_j\right)^2}\right] & i = 0, \quad 1 \leq j \leq n \\ \dfrac{2(-1)^{n+j}}{3d_j}\left[\dfrac{(2n^2+1)(1+x_j)-6}{\left(1+x_j\right)^2}\right] & i = n, \quad 0 \leq j \leq n-1 \\ \dfrac{(n^4-1)}{15} & i = j = \{0,n\} \end{cases} \tag{108}$$

The matrix $\boldsymbol{D}^{(2)}$ can be also approximated by a product of two matrices $\boldsymbol{D}^{(1)}$, i.e.,

$$\boldsymbol{D}^{(2)} \simeq \boldsymbol{D}^{(1)}\boldsymbol{D}^{(1)}, \tag{109}$$

although $\boldsymbol{D}^{(2)}$ is obviously more accurate than $\boldsymbol{D}^{(1)}\boldsymbol{D}^{(1)}$.

Appendix B: Gauss-Legendre interpolation and differentiation

The Legendre orthogonal polynomials $\{L_0(x), L_1(x), \dots, \}$ are defined in $[-1,1]$ by iterating the three-term recurrence relation

$$L_0(x) = 1, \tag{110}$$

$$L_1(x) = x, \tag{111}$$

$$L_{k+1}(x) = \frac{2k+1}{k+1} x L_k(x) - \frac{k}{k-1} L_{k-1}(x). \tag{112}$$

This yields, for example

$$L_2(x) = \frac{3}{2}x^2 - \frac{1}{2}, \qquad L_3(x) = \frac{5}{2}x^3 - \frac{3}{2}x, \qquad L_4(x) = \frac{35}{8}x^4 - \frac{15}{4}x^2 + \frac{3}{8}, \qquad \dots. \tag{113}$$

It can be shown that Legendre polynomials satisfy the orthogonality conditions

$$\int_{-1}^{1} L_k(x) L_j(x) dx = \delta_{kj} \left\| L_k \right\|^2_{L^2([-1,1])}. \tag{114}$$

where

$$\left\| L_k \right\|^2_{L^2([-1,1])} = \int_{-1}^{1} L_k^2(x) dx = \frac{2}{2n+1}. \tag{115}$$

In Table 3 we summarize the definition of the Gauss-Legendre (GL) and Gauss-Legendre-Lobatto (GLL) interpolation nodes, as well as the corresponding Lagrange characteristic polynomials. The GL and GLL first-order differentiation matrices, i.e.,

$$D^{(1)}_{ij} = \frac{dl_j(x_i)}{dx} \tag{116}$$

can be derived analytically as

$$D^{(1)}_{ij} = \begin{cases} \dfrac{L'_{n+1}(x_i)}{(x_i - x_j)L'_{n+1}(x_j)} & i \neq j \\ \dfrac{x_i}{(1-x_i)^2} & i = j \end{cases} \qquad \text{(Gauss-Legendre)}, \tag{117}$$

$$D^{(1)}_{ij} = \begin{cases} -\dfrac{n(n+1)}{4} & i = j = 0 \\ \dfrac{L_n(x_i)}{(x_i - x_j)L_n(x_j)} & i \neq j \\ 0 & i = j \quad \text{(not 0 or } n\text{)} \\ \dfrac{n(n+1)}{4} & i = j = n \end{cases} \qquad \text{(Gauss-Legendre-Lobatto)}. \tag{118}$$

	Gauss-Legendre (GL)	Gauss-Legendre-Lobatto (GLL)
nodes $\{x_0, \dots, x_n\}$	$L_{n+1}(x) = 0$	$(1 - x^2)L_n'(x) = 0$
Lagrange polynomials	$l_i(z) = \dfrac{L_{n+1}(z)}{(z - z_i)L_{n+1}'(z)}$	$l_i(z) = -\dfrac{1}{n(n+1)}\dfrac{(1 - x^2)}{(x - x_i)}\dfrac{L_n'(x)}{L_n'(x_i)}$

Table 3: Gauss-Legendre and Gauss-Lobatto-Legendre polynomial interpolation rules.

References

[1] J. S. Hesthaven, S. Gottlieb, and D. Gottlieb. *Spectral methods for time-dependent problems*, volume 21 of *Cambridge Monographs on Applied and Computational Mathematics.* Cambridge University Press, Cambridge, 2007.

[2] A. Quarteroni, R. Sacco, and F. Salieri. *Numerical mathematics.* Springer, 2007.