Let us consider bounded and nonperiodic domains such as an interval $[-1,1]$ on which a nonperiodic function $u(x)$ is defined.

A suitable basis set is polynomial basis $\{\phi_n(x)\}$, in particular $\{x^n\}$, provided only that $u(x)$ can be well approximated in the finite dimensional subspace

$$B_N = \text{span}\{x^n \mid n = 0,1,\ldots,N\}$$

by the projection operator in the continuous case:

$$(P_N u)(x) = \sum_{n=0}^{N} \hat{u}_n x^n,$$

or by the interpolation operator in the discrete case:

$$(I_N u)(x) = \sum_{n=0}^{N} \tilde{u}_n x^n$$

where the interpolation condition $(I_N u)(x_j) = u(x_j)$ is satisfied at the suitably chosen collocation points $\{x_j\}_{j=0}^{N}$.

- **<u>Chebyshev Polynomials</u>**

This is a suitable basis for B_N and a member of the ultraspherical Jacobi polynomial family $P_n^{(\alpha,\beta)}$ with $\alpha = \beta$ that minimizes the error in the maximum norm, $L^\infty[-1,1]$, i.e.

$$\min_{\hat{u}_n}\left[\max_{-1\le x\le 1}\left| u(x) - \sum_{n=0}^{N} \hat{u}_n \, P_n^{(\alpha,\beta)}(x)\right|\right] \quad \text{(minimax problem)},$$

the answer is the Chebyshev polynomials which are derived from the Jacobi polynomials with the choice of $\alpha = \beta = -1/2$.

They appear as eigensolutions to the singular S-L problem

$$\frac{d}{dx}\left(\sqrt{1-x^2}\,\frac{dT_n}{dx}\right) + \frac{n^2}{\sqrt{1-x^2}}\,T_n(x) = 0$$

where the weight function is $w(x) = 1/\sqrt{1-x^2})$ and the first few are

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_2(x) = 2x^2 - 1, \quad \cdots$$

with

$$T_n(x) = \cos\left(n\cos^{-1}(x)\right),$$

Continuous versus Discrete expansions

Consider the truncated continuous expansion of functions, $u(x) \in L^2_w[-1,1]$, in ultraspherical polynomials of the form

$$(P_N u)(x) \doteq \sum_{n=0}^{N} \hat{u}_n P_n^{(\alpha)}(x).$$

The expansion coefficients are found as

$$\hat{u}_n = \frac{1}{\gamma_n}\left(u\,,P_n^{(\alpha)}\right)_w \equiv \frac{1}{\gamma_n}\int_{-1}^{1} u(x)\,P_n^{(\alpha)}(x)\underbrace{\left(1-x^2\right)^{\alpha}}_{w(x)}dx$$

where $\gamma_n = \left(P_n^{(\alpha)}\,, P_n^{(\alpha)}\right)_w = \left\|P_n^{(\alpha)}\right\|_w^2$.

Thus, the application of the ultraspherical polynomials require evaluation of the expansion coefficients through the computation of an integral which is clearly not practical for the use in connection with computers.

Gaussian quadrature integration to the resque.

The numerical integration formula in the form:

$$\int_a^b f(x)w(x)dx \approx \sum_{i=0}^{n} c_i f(x_i)$$

is called a quadrature formula wrt positive weight $w(x)$ for the coefficients so chosen as

$$c_i = \int_a^b L_i(x)w(x)dx$$

with $L_i(x)$ polynomial cardinal functions (Lagrange interpolants) that the formula is exact for f polynomial of degree $\leq n$ for given nodes $\{x_0, x_1, \ldots, x_n\}$ in $[a,b]$.

This formula can be extended to produce exact result for the largest class of polynomials by providing freedom in selecting the nodes. Since there will be $2n+2$ parameters to choose, the class of polynomials of degree $\leq 2n+1$ will be the largest class of polynomials for which the formula is exact.

An insight to the construction procedure in the previous example is provided by Gauss' remarkable result:

Gaussian Quadrature Theorem: Let q be a nontrivial polynomial of degree $n+1$ such that

$$\int_a^b x^k q(x)w(x)dx = 0 \quad \text{for} \quad k = 0,1,\dots,n.$$

Let $\{x_0, x_1, \dots, x_n\}$ be the zeros of q. Then the formula

$$\int_a^b f(x)w(x)dx \approx \sum_{i=0}^n c_i f(x_i) \quad \text{where} \quad c_i = \int_a^b L_i(x)w(x)dx$$

with these x_i's as nodes will be exact for all polynomials of degree $\leq 2n+1$.

Let f be a polynomial of degree $2n+1$ and p be of degree n, then

$$f(x) = q(x)p(x) + r(x)$$

where the remainder $r(x)$ is also of degree n and $f(x_i) = r(x_i)$. Then

$$\int_a^b f(x)w(x)dx = \int_a^b r(x)w(x)dx = \sum_{i=0}^n c_i r(x_i) = \sum_{i=0}^n c_i f(x_i)$$

due to the fact that the quadrature formula is exact for polynomials of degree n.

Gauss-Lobatto Integration: Let $-1 = x_0 < x_1 < \cdots < x_N = 1$ be the $N+1$ roots of the polynomial $q(x) = P_{N+1}^{(\alpha)}(x) + a\,P_N^{(\alpha)}(x) + b\,P_{N-1}^{(\alpha)}(x)$, where a and b are chosen to produce $q(-1) = q(1) = 0$, and let $\omega_0, \omega_1, \dots, \omega_N$ be the solution of the linear system

$$\sum_{j=0}^N (x_j)^k \omega_j = \int_{-1}^1 x^k w(x)\,dx, \quad 0 \leq k \leq N.$$

Then,

$$\sum_{j=0}^N p(x_j)\omega_j = \int_{-1}^1 p(x)w(x)\,dx \quad \text{for all } p \in B_{2N-1}.$$

There is a more convenient representation of $q(x)$, through which we obtain the collocation points. The collocation points also appear as the roots of the polynomial

$$q(x) = (1-x^2)\frac{d}{dx}P_N^{(\alpha)}(x).$$

Quadrature for Chebyshev Polynomials

Chebyshev polynomials, besides the fact that they are well suited for approximation in the maximum norm, are distinguished for explicit and simple expressions for the quadrature points as well as the corresponding weights in the Chebyshev quadrature.

Chebyshev Gauss-Lobatto quadrature points appear as the roots of the polynomial

$$q(x) = T_{N+1}(x) - T_{N-1}(x) = -\frac{2}{N}(1-x^2)\frac{d}{dx}T_N(x),$$

yielding

$$x_j = -\cos\left(\frac{\pi}{N} j\right), \qquad j = 0,\ldots,N.$$

The corresponding weights are given as

$$\omega_j = \begin{cases} \pi/2N & j = 0, N \\ \pi/N & j = 1,\ldots,N-1 \end{cases}.$$

Discrete Inner product

The quadrature formulas suggest a definition of a discrete version of the inner product. Recall that in the continuous case we have

$$(f\,,\,g)_w \equiv \int_{-1}^{1} f(x)g(x)\,w(x)\,dx$$

for $f, g \in L^2_w[-1,1]$.

From the development of the quadrature formulas it is natural to define the discrete inner product

$$[f\,,\,g]_\omega \equiv \sum_{j=0}^{N} f(x_j)\,g(x_j)\,\omega_j$$

where x_j can be any of the Gauss quadrature points with the corresponding weights ω_j and $f, g \in B_N$. Note that

$$[f\,,\,g]_\omega = (f\,,\,g)_w$$

in the case of Gauss-Radau and Gauss quadratures, while this is not the case when using the Gauss-Lobatto quadrature since $f(x)g(x) \in B_{2N}$ for which the quadrature fails to be exact. This only affects the computation of the norm

$$\gamma_N = \left[P_N^{(\alpha)}\,,\,P_N^{(\alpha)}\right]_\omega \equiv \left\|P_N^{(\alpha)}\right\|_\omega$$

over the Gauss-Lobatto quadrature points which can be computed by other means.

The Discrete Chebyshev Expansion

For the specific case of the Chebyshev polynomials, $P_n^{(-1/2)}(x) \equiv T_n(x)$, the discrete orthogonality relation is given by

$$[T_n\ ,\ T_m]_\omega \equiv \sum_{j=0}^{N} T_n(x_j)\, T_m(x_j)\, \omega_j = \gamma_n \delta_{nm}$$

where the discrete norms are

$$\gamma_n = [T_n\ ,\ T_n]_\omega = \begin{cases} c_n \pi/2 & n < N \\ \pi/2 & n = N \ \text{ for Gauss and Gauss-Radau} \\ \pi & n = N \ \text{ for Gauss-Lobatto} \end{cases}$$

with $c_0 = 2$ and $c_{n>0} = 1$.

The development of the quadrature rules and in turn the discrete orthogonality relation facilitates now to devise accurate methods for the computation of the expansion coefficients in the discrete expansion in Chebyshev polynomials of the form

$$(I_N u)(x) = \sum_{n=0}^{N} \tilde{u}_n T_n(x)\ , \qquad \tilde{u}_n = \frac{1}{\gamma_n} \sum_{j=0}^{N} u(x_j) T_n(x_j)\, \omega_j$$

where $(I_N u)(x)$ interpolates $u(x)$ exactly at the quadrature points, i.e. $(I_N u)(x_j) = u(x_j)$.

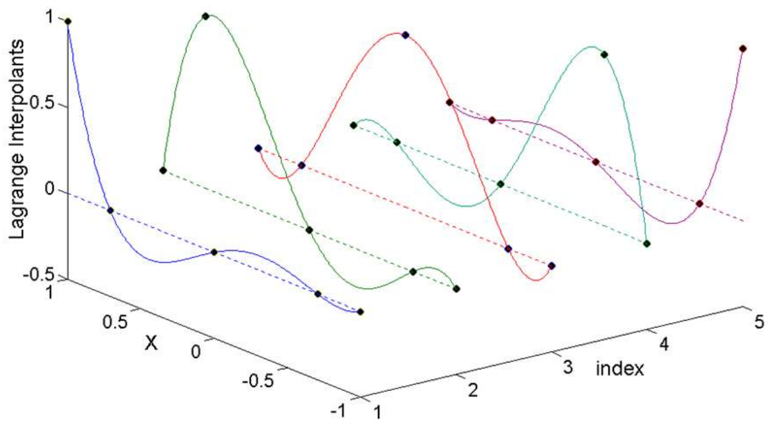
In a different formulation we recover

$$\begin{aligned} (I_N u)(x) &= \sum_{n=0}^{N} \tilde{u}_n T_n(x) \\ &= \sum_{n=0}^{N} \left(\frac{1}{\gamma_n} \sum_{j=0}^{N} u(x_j) T_n(x_j)\, \omega_j \right) T_n(x) \\ &= \sum_{j=0}^{N} u(x_j) \underbrace{\left(\omega_j \sum_{n=0}^{N} \frac{1}{\gamma_n}\, T_n(x)\, T_n(x_j) \right)}_{L_j(x)} \end{aligned}$$

that follows from *Christoffel-Darboux identity*. Here we define the polynomial, $L_j(x) \in B_N$, as the interpolating Lagrange polynomial based on the quadrature nodes $\{x_j\}_{j=0}^{N}$ such that $L_j(x_k) = \delta_{jk}$.

Another explicit formula for $L_j(x)$ is

$$L_j(x) = \prod_{\substack{k=0 \\ k \neq j}}^{N} \frac{(x - x_k)}{(x_j - x_k)}.$$



Lagrange Interpolation*

The classical Lagrange form of polynomial interpolation

$$p(x) = \sum_{j=0}^{n} f_j L_j(x), \qquad L_j(x) = \prod_{k=0,\,k\neq j}^{n}(x - x_k) \Big/ \prod_{k=0,\,k\neq j}^{n}(x_j - x_k)$$

with $L_j(x_k) = \delta_{jk}$, has certain shortcomings for practical computations:

1. Each evaluation of $p(x)$ requires $O(n^2)$ additions and multiplications (flops).
2. Adding a new data pair x_{n+1}, f_{n+1} requires a new computation from scratch.
3. The computation is numerically unstable.

For this purpose, Newton form is generally recommended which require only $O(n)$ flops for each evaluation of p.

The Lagrange formula can be written in such a way that it too can be evaluated and updated in $O(n)$ operations. This is called barycentric form of Lagrange interpolation.

* J.P.Berrut and L.N.Trefethen, Barycentric Lagrange interpolation, SIAM Rev., 2004.

Chebyshev Differentiation Matrices

By using the alternative (nodal) formulation of the discrete expansion in interpolating Lagrange polynomials, the computation of derivative of u(x) at the quadrature (collocation) points is accomplished in the same way as before

$$I_N \frac{d}{dx}(I_N u)\Big|_{x=x_k} = \sum_{j=0}^{N} u(x_j) \frac{dL_j}{dx}\Big|_{x=x_k} \equiv \sum_{j=0}^{N} D_{kj}\, u(x_j)$$

where D denotes the $(N+1)\times(N+1)$ differentiation matrix.

For the most popular case of the Gauss-Lobatto quadrature points, $x_j = -\cos(\pi j/N)$, the entries of D are given by

$$D = \begin{bmatrix} D_{00} = \frac{2N^2+1}{6} & \cdots & D_{ij} = \frac{c_i}{c_j}\frac{(-1)^{i+j}}{(x_i - x_j)}\Big|_{\substack{i,j=0\\ i\neq j}}^{N} \\ \vdots & D_{jj} = -\frac{x_j}{2(1-x_j^2)}\Big|_{j=1}^{N-1} & \vdots \\ D_{ij} = \frac{c_i}{c_j}\frac{(-1)^{i+j}}{(x_i - x_j)}\Big|_{\substack{i,j=0\\ i\neq j}}^{N} & \cdots & D_{NN} = -\frac{2N^2+1}{6} \end{bmatrix}$$

which can be obtained using $\omega_j = (-1)^j \delta_j$ in the Barycentric form of the polynomial differentiation.

The Matlab function **cheb.m** returns a vector, **x**, the vector of the Gauss-Lobatto points and a matrix, **D**, the Chebyshev differentiation matrix for a given N.

```
% cheb : compute D, differentiation matrix, x,Chebyshev grid.

  function [D,x] = cheb(N)
  if N==0, D=0; x=1; return, end
  x = cos(pi*(0:N)/N)';
  c = [2; ones(N-1,1); 2].*(-1).^(0:N)';
  X = repmat(x,1,N+1);
  dX = X-X';
  D  = (c*(1./c)')./(dX+(eye(N+1)));  % off-diagonal entries
  D  = D - diag(sum(D'));             % diagonal entries
```

Matlab Syntax

B = repmat(A,m,n) creates a large matrix B consisting of an m-by-n tiling of copies of A.
e.g. B = repmat(eye(2),1,3)

$$B = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

Note : Function **cheb** features a indirect method of computing the diagonal elements of D that is found to produce a matrix with better stability properties in the presence of rounding errors and thus yields more accurate results than the direct formulas above. The indirect method uses the identity

$$D_{ii} = -\sum_{j=0,j\neq i}^{N} D_{ij}$$

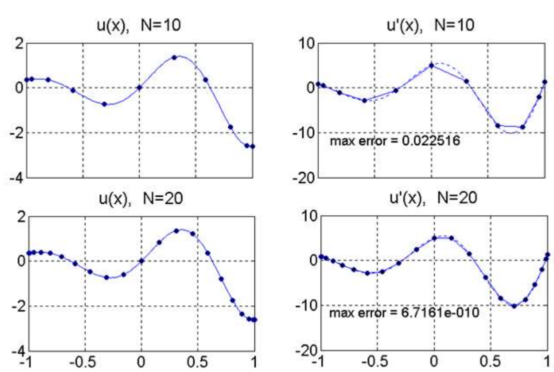
to obtain the diagonal entries from the off-diagonal entries computed by the use of direct formulas.

This identity follows from the observation that the polynomial interpolant to the discrete function $[1 \quad 1 \quad \cdots \quad 1]^T$ is the constant function $(I_N u)(x) = 1$ and since $(d/dx)(1) = 0$, D must map $[1 \quad 1 \quad \cdots \quad 1]^T$ to the zero vector.

Example : **The m-file, p11.m, illustrates how D from cheb.m can be used to differentiate the smooth nonperiodic function**

$$\mathbf{u(x) = exp(x)sin(5x)}$$

on Gauss-Lobatto grids with $N = 10$ and $N = 20$.



Note that the error is drastically reduced by doubling N. With $N = 20$, we get nine-digit accuracy.

The computation of higher derivatives directly follows the procedure for the computation of the first derivative, namely,

$$\frac{d^\nu}{dx^\nu}(I_N u)(x) = \sum_{j=0}^{N} u(x_j)\frac{d^\nu L_j(x)}{dx^\nu}.$$

Alternatively, one may compute the ν-th order differentiation matrix by simply multiplying the first order differentiation matrix, i.e.

$$D^{(\nu)} = (D)^\nu.$$

Although this latter approach is most straightforward, one should always use the exact expressions for the entries of the operators, if available, due to better numerical properties, i.e. lesser effect of round-off error.

Properties of D

1. The Gauss-Lobatto differentiation matrix D of size $(N+1)\times(N+1)$ is nilpotent, i.e. D^k is the zero matrix for some positive integer k. In fact, D^{N+1} is identically zero.
2. It is centro-antisymmetric : $D_{ij} = -D_{N-i,N-j}$.
3. Rank of D is N. Similarly, rank of $D^{(2)} = D^2$ is $N-1$ and further, rank of $D^{(\nu)}$ is $N-(\nu-1)$.

This result implies that the ν-th order Chebyshev differentiation should only be applied at $N-(\nu-1)$ grid points, i.e. ν of $(N+1)$ grid points should be specified, otherwise the resulting discretization matrix will be singular. This actually is in analogy with the definition of a well-posed problem that the ν-th order differential equation requires ν initial or boundary conditions, which provide ν equations.