Numerical Methods I
Orthogonal Polynomials

Aleksandar Donev
Courant Institute, NYU[1]
donev@courant.nyu.edu

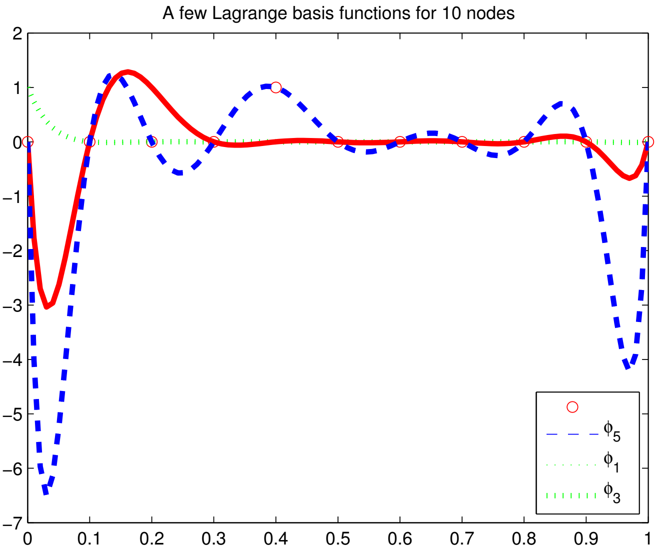
[1]MATH-GA 2011.003 / CSCI-GA 2945.003, Fall 2014

Nov 6th, 2014

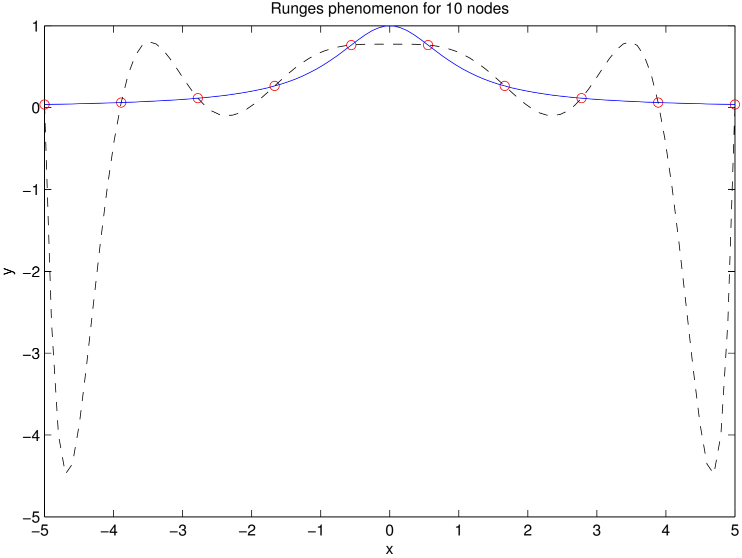
Outline

1. Review
2. Orthogonal Polynomials on $[-1, 1]$
3. Spectral Approximation
4. Conclusions

Lagrange basis on 10 nodes



Runge's phenomenon $f(x) = (1 + x^2)^{-1}$



Function Space Basis

- Think of a function as a vector of coefficients in terms of a set of n **basis functions**:
$$\{\phi_0(x), \phi_1(x), \ldots, \phi_n(x)\},$$
for example, the monomial basis $\phi_k(x) = x^k$ for polynomials.
- A finite-dimensional approximation to a given function $f(x)$:
$$\tilde{f}(x) = \sum_{i=1}^{n} c_i \phi_i(x)$$
- **Least-squares approximation** for $m > n$ (usually $m \gg n$):
$$\mathbf{c}^\star = \arg\min_{\mathbf{c}} \left\| f(x) - \tilde{f}(x) \right\|_2,$$
which gives the **orthogonal projection** of $f(x)$ onto the finite-dimensional basis.

Least-Squares Fitting

- Discrete case: Think of **fitting** a straight line or quadratic through experimental data points.
- The function becomes the vector $\mathbf{y} = \mathbf{f}_{\mathcal{X}}$, and the approximation is

$$y_i = \sum_{j=1}^{n} c_j \phi_j(x_i) \quad \Rightarrow \quad \mathbf{y} = \mathbf{\Phi} \mathbf{c},$$

$$\mathbf{\Phi}_{ij} = \phi_j(x_i).$$

- This means that finding the approximation consists of solving an **overdetermined linear system**

$$\mathbf{\Phi} \mathbf{c} = \mathbf{y}$$

- Note that for $m = n$ this is equivalent to interpolation. MATLAB's *polyfit* works for $m \geq n$.

Normal Equations

- Recall that one way to solve this is via the normal equations:

$$\left(\boldsymbol{\Phi}^{\star}\boldsymbol{\Phi}\right)\mathbf{c}^{\star}=\boldsymbol{\Phi}^{\star}\mathbf{y}$$

- A basis set is an **orthonormal basis** if

$$(\phi_i,\phi_j)=\sum_{k=0}^{m}\phi_i(x_k)\phi_j(x_k)=\delta_{ij}=\begin{cases}1 & \text{if } i=j\\ 0 & \text{if } i\neq j\end{cases}$$

$$\boldsymbol{\Phi}^{\star}\boldsymbol{\Phi}=\mathbf{I}\text{ (unitary or orthogonal matrix)}\quad\Rightarrow$$

$$\mathbf{c}^{\star}=\boldsymbol{\Phi}^{\star}\mathbf{y}\quad\Rightarrow\quad c_i=\phi_i^{\mathcal{X}}\cdot\mathbf{f}_{\mathcal{X}}=\sum_{k=0}^{m}f(x_k)\phi_i(x_k)$$

Orthogonal Polynomials

- Consider a function on the interval $I = [a, b]$.
  Any finite interval can be transformed to $I = [-1, 1]$ by a simple transformation.
- Using a **weight function** $w(x)$, define a **function dot product** as:

$$(f, g) = \int_a^b w(x) \left[f(x) g(x) \right] dx$$

- For different choices of the weight $w(x)$, one can explicitly construct **basis of orthogonal polynomials** where $\phi_k(x)$ is a polynomial of degree k (**triangular basis**):

$$(\phi_i, \phi_j) = \int_a^b w(x) \left[\phi_i(x) \phi_j(x) \right] dx = \delta_{ij} \left\| \phi_i \right\|^2 .$$

Legendre Polynomials

- For equal weighting $w(x) = 1$, the resulting triangular family of of polynomials are called **Legendre polynomials**:

$$\phi_0(x) = 1$$

$$\phi_1(x) = x$$

$$\phi_2(x) = \frac{1}{2}(3x^2 - 1)$$

$$\phi_3(x) = \frac{1}{2}(5x^3 - 3x)$$

$$\phi_{k+1}(x) = \frac{2k+1}{k+1} x\phi_k(x) - \frac{k}{k+1}\phi_{k-1}(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}\left[\left(x^2 - 1\right)^n\right]$$

- These are orthogonal on $I = [-1, 1]$:

$$\int_{-1}^{-1} \phi_i(x)\phi_j(x)dx = \delta_{ij} \cdot \frac{2}{2i+1}.$$

Interpolation using Orthogonal Polynomials

- Let's look at the **interpolating polynomial** $\phi(x)$ of a function $f(x)$ on a set of $m+1$ **nodes** $\{x_0, \ldots, x_m\} \in I$, expressed in an orthogonal basis:

$$\phi(x) = \sum_{i=0}^{m} a_i \phi_i(x)$$

- Due to orthogonality, taking a dot product with ϕ_j (**weak formulation**):

$$(\phi, \phi_j) = \sum_{i=0}^{m} a_i (\phi_i, \phi_j) = \sum_{i=0}^{m} a_i \delta_{ij} \|\phi_i\|^2 = a_j \|\phi_j\|^2$$

- This is **equivalent to normal equations** if we use the right dot product:

$$(\mathbf{\Phi}^\star \mathbf{\Phi})_{ij} = (\phi_i, \phi_j) = \delta_{ij} \|\phi_i\|^2 \text{ and } \mathbf{\Phi}^\star \mathbf{y} = (\phi, \phi_j)$$

Gauss Integration

$$a_j \left\|\phi_j\right\|^2 = (\phi, \phi_j) \quad \Rightarrow \quad a_j = \left(\left\|\phi_j\right\|^2\right)^{-1} (\phi, \phi_j)$$

- Question: Can we easily compute

$$(\phi, \phi_j) = \int_a^b w(x) \left[\phi(x)\phi_j(x)\right] dx = \int_a^b w(x) p_{2m}(x) dx$$

  for a polynomial $p_{2m}(x) = \phi(x)\phi_j(x)$ of degree at most $2m$?
- Let's first consider polynomials of degree at most m

$$\int_a^b w(x) p_m(x) dx = ?$$

Gauss Weights

- Now consider the **Lagrange basis** $\{\varphi_0(x), \varphi_1(x), \ldots, \varphi_m(x)\}$, where you recall that
$$\varphi_i(x_j) = \delta_{ij}.$$
- Any polynomial $p_m(x)$ of degree at most m can be expressed in the Lagrange basis:
$$p_m(x) = \sum_{i=0}^{m} p_m(x_i)\varphi_i(x),$$
$$\int_a^b w(x)p_m(x)dx = \sum_{i=0}^{m} p_m(x_i)\left[\int_a^b w(x)\varphi_i(x)dx\right] = \sum_{i=0}^{m} w_i p_m(x_i),$$
where the **Gauss weights w** are given by
$$w_i = \int_a^b w(x)\varphi_i(x)dx.$$

Back to Interpolation

- For any polynomial $p_{2m}(x)$ there exists a polynomial quotient q_{m-1} and a remainder r_m such that:

$$p_{2m}(x) = \phi_{m+1}(x)q_{m-1}(x) + r_m(x)$$

$$\int_a^b w(x)p_{2m}(x)dx = \int_a^b \left[w(x)\phi_{m+1}(x)q_{m-1}(x) + w(x)r_m(x)\right]dx$$
$$= (\phi_{m+1}, q_{m-1}) + \int_a^b w(x)r_m(x)dx$$

- But, since $\phi_{m+1}(x)$ is orthogonal to any polynomial of degree at most m, $(\phi_{m+1}, q_{m-1}) = 0$ and we thus get:

$$\int_a^b w(x)p_{2m}(x)dx = \sum_{i=0}^{m} w_i r_m(x_i)$$

Gauss nodes

- Finally, if we choose the **nodes to be zeros of** $\phi_{m+1}(x)$, then

$$r_m(x_i) = p_{2m}(x_i) - \phi_{m+1}(x_i)q_{m-1}(x_i) = p_{2m}(x_i)$$

$$\int_a^b w(x)p_{2m}(x)dx = \sum_{i=0}^{m} w_i p_{2m}(x_i)$$

and thus we have found a way to **quickly project any polynomial** of degree up to m onto the basis of orthogonal polynomials:

$$(p_m, \phi_j) = \sum_{i=0}^{m} w_i p_m(x_i)\phi_j(x_i)$$

Interpolation using Gauss weights

- Recall we want to compute

$$a_j = \left(\|\phi_j\|^2\right)^{-1}(\phi, \phi_j),$$

where for Legendre polynomials

$$\|\phi_j\|^2 = \frac{2}{2j+1}.$$

- Now we know how to compute fast:

$$(\phi, \phi_j) = \sum_{i=0}^{m} w_i \phi(x_i)\phi_j(x_i) = \sum_{i=0}^{m} w_i f(x_i)\phi_j(x_i) \quad \Rightarrow$$

$$a_j = \left(\frac{2j+1}{2}\right)\sum_{i=0}^{m} w_i f(x_i)\phi_j(x_i),$$

where w_i and $\phi_j(x_i)$ are coefficients that can be **precomputed** and **tabulated**.

Gauss-Legendre polynomials

- For any weighting function the polynomial $\phi_k(x)$ has k simple zeros all of which are in $(-1, 1)$, called the (order k) **Gauss-Legendre nodes**, $\phi_{m+1}(x_i) = 0$.
- The interpolating polynomial $\phi(x_i) = f(x_i)$ on the Gauss nodes is the **Gauss-Legendre interpolant** $\phi_{GL}(x)$.
- The orthogonality relation can be expressed as a **sum instead of integral**:

$$(\phi_i, \phi_j) = \sum_{i=0}^{m} w_i \phi_i(x_i) \phi_j(x_i) = \delta_{ij} \|\phi_i\|^2$$

- We can thus define a new weighted **discrete dot product**

$$\mathbf{f} \cdot \mathbf{g} = \sum_{i=0}^{m} w_i f_i g_i$$

Discrete Orthogonality of Polynomials

- The orthogonal polynomial basis is **discretely-orthogonal** in the new dot product,

$$\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}_j = (\phi_i, \phi_j) = \delta_{ij} \left(\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}_i \right)$$

- This means that the matrix in the normal equations is diagonal:

$$\boldsymbol{\Phi}^{\star} \boldsymbol{\Phi} = \text{Diag} \left\{ \|\phi_0\|^2, \dots, \|\phi_m\|^2 \right\} \quad \Rightarrow \quad a_i = \frac{\mathbf{f} \cdot \boldsymbol{\phi}_i}{\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}_i}.$$

- The Gauss-Legendre interpolant is thus easy to compute:

$$\phi_{GL}(x) = \sum_{i=0}^{m} \frac{\mathbf{f} \cdot \boldsymbol{\phi}_i}{\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}_i} \phi_i(x).$$

Chebyshev Interpolation

- There are other **families of orthogonal polynomials** that are also very useful in practice (Gauss-Lobato, Gauss-Hermite, etc.).
- A notable example are the **Chebyshev polynomials** on $[-1, 1]$, with weight function

$$w(x) = \frac{1}{\sqrt{1 - x^2}}$$

defined recursively via:

$$\phi_0 = 1$$
$$\phi_1 = x$$
$$\phi_{k+1} = 2x\phi_k - \phi_{k-1}.$$

- **Orthogonality** relation, for i, j not both zero,

$$(\phi_i, \phi_j) = \delta_{ij}\frac{\pi}{2} \text{ and } (\phi_0, \phi_0) = \pi.$$

Chebyshev Nodes

- They can also be defined as the unique polynomials satisfying the **trig relation**:

$$\phi_k(\cos\theta) = \cos(k\theta).$$

- This means that their roots $\phi_k(x_i) = 0$, the **Chebyshev nodes**, are easy to find,

$$x_i = \cos\left(\frac{2i-1}{2k}\pi\right), \quad i = 1, \ldots, k,$$

  which have a simple geometric interpretation as the projection of uniformly spaced points on the unit circle.

- The **Chebyshev-Gauss weights** are also easy to compute,

$$w_i = \int_a^b w(x)\varphi_i(x)dx = \frac{\pi}{k}.$$

- Polynomial interpolation using the Chebyshev nodes **eliminates Runge's phenomenon**.

Hilbert Space L^2_w

- Consider the **Hilbert space** L^2_w of **square-integrable functions** on $[-1,1]$:

$$\forall f \in L^2_w : \quad (f,f) = \|f\|^2 = \int_{-1}^{1} w(x) \left[f(x) \right]^2 dx < \infty.$$

- **Legendre polynomials** form a **complete orthogonal basis** for L^2_w:

$$\forall f \in L^2_w : \quad f(x) = \sum_{i=0}^{\infty} f_i \phi_i(x)$$

$$f_i = \frac{(f,\phi_i)}{(\phi_i,\phi_i)}.$$

- The least-squares approximation of f is a **spectral approximation** and is obtained by simply truncating the infinite series:

$$\phi_{sp}(x) = \sum_{i=0}^{m} f_i \phi_i(x).$$

Spectral approximation

Continuous (spectral approximation): $\phi_{sp}(x) = \sum_{i=0}^{m} \frac{(f, \phi_i)}{(\phi_i, \phi_i)} \phi_i(x).$

Discrete (interpolating polynomial): $\phi_{GL}(x) = \sum_{i=0}^{m} \frac{\mathbf{f} \cdot \boldsymbol{\phi}_i}{\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}_i} \phi_i(x).$

- If we **approximate** the function dot-products with the discrete weighted products

$$(f, \phi_i) \approx \sum_{j=0}^{m} w_j f(x_j) \phi_i(x_j) = \mathbf{f} \cdot \boldsymbol{\phi}_i,$$

we see that the Gauss-Legendre interpolant is a **discrete spectral approximation**:

$$\phi_{GL}(x) \approx \phi_{sp}(x).$$

Discrete spectral approximation

- Using a spectral representation has many advantages for function approximation: **stability**, **rapid convergence**, easy to **add more basis functions**.
- The convergence, for sufficiently smooth (nice) functions, is **more rapid than any power law**

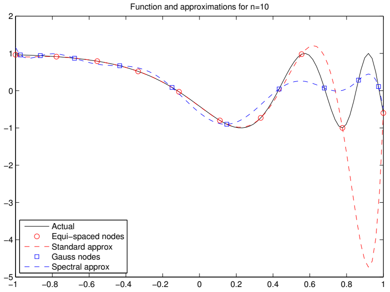
$$\|f(x) - \phi_{GL}(x)\| \leq \frac{C}{N^d} \left(\sum_{k=0}^{d} \left\| f^{(k)} \right\|^2 \right)^{1/2},$$

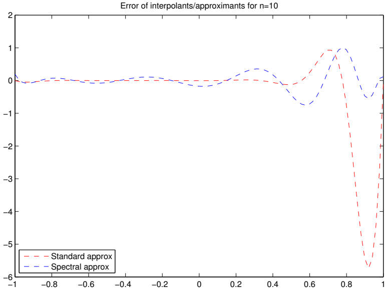
  where the multiplier is related to the **Sobolev norm** of $f(x)$.
- For $f(x) \in \mathcal{C}^1$, the convergence is also **pointwise** with similar accuracy ($N^{d-1/2}$ in the denominator).
- This so-called **spectral accuracy** (limited by smoothness only) cannot be achived by piecewise, i.e., local, approximations (limited by order of local approximation).

Regular grids

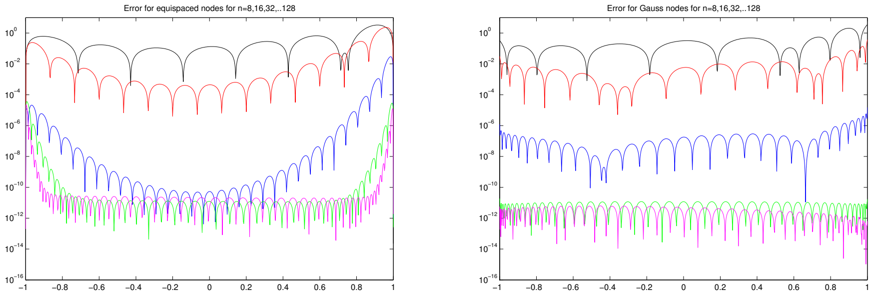
```
a=2;
f = @(x) cos(2*exp(a*x));

x_fine=linspace(-1,1,100);
y_fine=f(x_fine);

% Equi-spaced nodes:
n=10;
x=linspace(-1,1,n);
y=f(x);
c=polyfit(x,y,n);
y_interp=polyval(c,x_fine);

% Gauss nodes:
[x,w]=GLNodeWt(n); % See webpage for code
y=f(x);
c=polyfit(x,y,n);
y_interp=polyval(c,x_fine);
```

Gauss-Legendre Interpolation

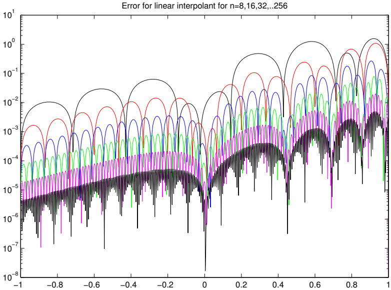


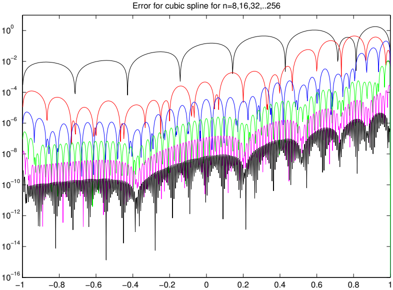


Global polynomial interpolation error



Local polynomial interpolation error





Conclusions/Summary

- Once a function dot product is defined, one can construct **orthogonal basis** for the space of functions of finite $2-$norm.
- For functions on the interval $[-1, 1]$, **triangular families of orthogonal polynomials** $\phi_i(x)$ provide such a basis, e.g., **Legendre** or **Chebyshev** polynomials.
- If one discretizes at the **Gauss nodes**, i.e., the roots of the polynomial $\phi_{m+1}(x)$, and defines a suitable **discrete Gauss-weighted dot product**, one obtains **discretely-orthogonal** basis suitable for numerical computations.
- The interpolating polynomial on the Gauss nodes is closely related to the **spectral approximation** of a function.
- **Spectral convergence** is faster than any power law of the number of nodes and is only limited by the **global** smoothness of the function, unlike piecewise polynomial approximations limited by the choice of **local** basis functions.
- One can also consider **piecewise-spectral approximations**.