IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 21, Issue 5, Ser. 2 (Sept. – Oct. 2025), PP 11-28
www.iosrjournals.org

Orthogonal Polynomials And Least Square Sense Approximation Of Data And Complex Functions.

V. Vijayakumar,
B-111, Saivihar, Sector-15, CBD-Belapur, Navimumbai-400614, India.

Abstract
The orthogonal polynomial set's properties and its use in the least square sense approximation of data or complex functions to a polynomial, are discussed. Performance of Chebyshev, Gram and Alfredo-Giuseppe (A-G) polynomials is examined with a simulated Planck profile with and without noise. For Gram and A-G orthogonal sets with equal spaced grids, the effect of two types of fake data points at the boundaries are examined by computing the root mean square deviation of the data fit. When three or more data points of zero values are added at fake grids, computation orthogonal polynomials of highest allowed degree are possible without divergence (no Runge phenomenon). When fake repeated data points are added, good data fit to polynomials (computed from the orthogonal polynomial for each degree) with highest convergence rate results. Good noise discrimination is seen in the latter case and can be used to identify the best orthogonal polynomial degree to be employed for data fitting applications.
Key Words: *Orthogonal polynomials, Function Approximation, Data fit*

Date of Submission: 01-10-2025 Date of Acceptance: 11-10-2025

I. Introduction

Orthogonal polynomial set (defined over an appropriate variable domain and boundary conditions, and a weighted orthogonality integral) are eigenvectors of certain classes of differential / difference operators [1]. This and the fact that they can be generated via recursive, differential or integral formula, have hypergeometric representation and associated generating functions, have close relation to continued fractions etc., endows it with several useful properties and consequent computational and analytical advantages. Orthogonal polynomial sets, by serving as a basis for expansion of arbitrary functions, [2] have applications in approximation theory and numerical analysis, as in the iteration free least square polynomial fitting, quantum mechanics, probability, signal & image analysis, etc. Here, after a concise discussion that traces the origin of the several useful properties of the orthogonal polynomials, its application in data fit is probed. The developments in orthogonal polynomials have a very long history and diversity with copious literature. However, here only references that are easily accessible (except for books), broader than the context, but with some bias towards data analysis, are quoted and that too disregarding the chronological order. An attempt is made to include all relevant details in a coherent manner.

II. Polynomial Sets And Finite (Converging) Hypergeometric Series

Polynomials offer unparalleled computational and analytical advantages and hypergeometric series [3a-c] is the general way to generate it with exemplary properties. The general hypergeometric series with the symbol ${}_pF_q\left|\begin{matrix}\alpha_1 & \alpha_2 & \cdots & \alpha_p\\ \delta_1 & \delta_2 & \cdots & \delta_q\end{matrix};z\right|$ or ${}_pF_q(\alpha_1,\alpha_2,\ldots,\alpha_p;\delta_1,\delta_2,\ldots,\delta_q;z)$ (variables separated by "," can be interchanged while that by ";" should not be) is defined in terms of shifted factorials (Pochhammer symbol, $(\alpha)_m = \prod_{i=1}^{m}[\alpha+i-1]$) as:- ${}_pF_q=\sum_{m=0}^{\infty}\frac{\prod_{i=1}^{p}(\alpha_i)_m}{\prod_{j=1}^{q}(\delta_j)_m}\frac{z^m}{m!}$. In terms of Gama function ($\Gamma(z)=\int_l^{\infty} t^{z-1}\,e^{-1}dt$), $(\alpha)_i=\frac{\Gamma(\alpha+i)}{\Gamma(i)}$ (because $\Gamma(i+1)=i\Gamma(i)=i!$), its definition is extended to non integer values of the parameters α and δ. Among the many interesting properties of the series, the following are of interest here:-

a) None of the δ_i can be negative or zero

b) The series is a n^{th} degree polynomial if anyone the numerators (α_i) is -n and hence can be used to generate a polynomial set of degree zero to n. The coefficients of such polynomials being function of factorials, represents various types probability distributions.

c) The convergence of the series depends on the ratio coefficients of the adjacent terms and the value of the variable, z. Thus convergence is obtained only for specific variable range; for example if p≤q, it converges for all values of z; b); if p=q+1 it converges for $|z|<1$ only ; and c)if $p > q + 1$ then the ratio of coefficients grows without bound unless z=o. However it can be a polynomial.

Particular cases of hypergeometric series corresponds to functions expressible as a polynomial series when an appropriate variable range is employed. A special case of interest here is ${}_2F_1(\alpha_1,\alpha_2;\delta_1;z)=\sum_{m=0}^{\infty}\frac{(\alpha_1)_m(\alpha_2)_m}{(\delta_1)_m m!}z^m = \sum_{m=0}^{\infty} r_m\, z^m = \frac{\Gamma(\delta_1)}{\Gamma(\alpha_1)\Gamma(\alpha_2)}\ \sum_{m=0}^{\infty}\frac{\Gamma(m+\alpha_1)\Gamma(m+\alpha_2)}{\Gamma(m+\delta_1)}\frac{z^m}{m!}$ which is a n^{th} degree polynomial when $(\delta_1-\alpha_1-\alpha_2)>0$ with α_1 or α_2 equal to -n.

d) The ratio of coefficients of two adjacent powers of z ($\frac{(\alpha_1+m)(\alpha_2+m)}{(\delta_1+m)(m+1)}$) is a characteristics of the hypergeometric series and it is a rational function of m. It's derivatives satisfy the relation $\frac{d^j}{dz^j}F(\alpha_1,\alpha_2;d_1;z) = \frac{(\alpha_1)_j(\alpha_2)_j}{(\delta)_j}F(\alpha_1+j,\alpha_2+j;d_1+j;z)$ linking hypergeometic series to the differential equation $z(1-z)y^{ll} + [\delta_1-(\alpha_1+\alpha_2+1)z]y^l - \alpha_1\,\alpha_2 y = 0$, with $y_n = {}_2F_1(-n,\alpha_2;\delta_1;z)$ (Jacobi polynomials) as a particular set of solutions. Other related system of differential equations, a) $zy^{ll} + [\ \delta-z]\ y^l - \alpha\ y=0$, with $y_n = {}_1F_1(\alpha;\delta;z)$ and b) $y''-2zy'+2ny=0$, with $y_n = (2z)^n{}_2F_0\left(\frac{-n}{2},\frac{-n(n-1)}{2};-;-z^{-2}\right)$ also have hypergeometric type polynomial sets as solution. The three linear ordinary differential equations of the type above that can be mapped onto each other and also reducible to a conjugate form by an appropriate transformation (transformation of abscissa, expressing ordinate as product of two functions or gauge transformation [4]) are referred to as canonical.

Polynomial representation for Data and Function.

Any differentiable function (f(x)) in an independent variable x or its function, with certain restriction about the point of expansion and the range (limited to a region of no singularity), can always be approximated to a converging polynomial within a characterize-able accuracy (Weierstrass's theorem) [5]; Taylor [6a], trigonometric, [6b] hyperbolic series [6c] (among many others) being such examples. This forms the basis for seeking polynomial as the solution of a large class of differential / difference equations and polynomial representation of complex functions and measured data. A n^{th} degree polynomial representation require n+1 linearly independent, n dependent basis sets [2]; the sets $\{x^j\}$, $\{p_n\}$ (polynomials of degree from zero to n), and cos(nx) (because cos(nx) can be expressed as a polynomial in cos(jx) with j=0 to n) being examples.

Polynomial approximation of data

Given a cluster of data points, to extract any insight, one needs to know the equation connecting an independent (in the simplest case of one dimension) variable, x, to the data as a dependent variable, y, (given by a function y=f(x)). If such a function is known with adjustable parameters, one may proceed with various optimization/regression methods [7a-d] and forms one of the approaches to data analysis. If no such model function is available or if such a function has a complex form, having a polynomial representation enables efficient computations.

With (N+1) discrete data (x_j,y_j) generated using a complex function or some measurement, it is always possible to construct a N degree polynomial that will have the value y_j at each x_j [8]. A $(N+1)^{th}$ degree polynomial, L_{N+1}, $[L_{N+1} = (\prod_{i=1}^{N+1}(x-x_i))]$ and its derivative at each grid point j, l^j, ($l^j = \prod_{i=0}^{N+1}(x_j - x_i)\ j \neq i$) can be used for constructing the N degree polynomial. Since $\frac{L(x)}{l^i(x-x_j)} = \delta_{ij}$, the N degree polynomial, p_N, [8] passing through all the points will have the form:-

$p_N = \sum_{i=1}^{N+1}\frac{L_{N+1}}{l^i(x-x_j)}y_j$ -------------------------------{1}.

An alternate way to approximate to a polynomial is to use the x^j basis and expand the N^{th} degree polynomial, p_N, as $p_N = \sum_{i=0}^{N} r_{N,j}\, x^j$ and obtain $r_{N,j}$ from the matrix inversion of the NxN Vandermonde (V) determinant [9] :-

$$\begin{vmatrix} y_0 \\ y_1 \\ . \\ y_N \end{vmatrix} = \begin{vmatrix} 1 & x_1 & . & x_1^N \\ 1 & x_2 & . & x_2^N \\ . & . & . & . \\ 1 & x_N & . & x_N^N \end{vmatrix}\begin{vmatrix} r_{N,0} \\ r_{N,1} \\ . \\ r_{N,N} \end{vmatrix}; ieF = VR \quad \text{---------------------------}\{2a\}$$

Inverting V [9] yield R as $V^{-1}F$.

If the above equation is multiplied by V^T this takes the form:-

$$\begin{vmatrix} \sum_0^N y_i \\ \sum_0^N x_i y_i \\ . \\ \sum_0^N {x_i}^N y_i \end{vmatrix} = \begin{vmatrix} N+1 & \sum_0^N x_i & . & \sum_0^N x_i^{N+1} \\ \sum_0^N x_i & \sum_0^N {x_i}^2 & . & \sum_0^N x_i^{N+2} \\ . & . & . & . \\ \sum_0^N {x_i}^{N+1} & \sum_0^N x_i^{N+2} & . & \sum_0^N x_i^{2(N+1)} \end{vmatrix}\begin{vmatrix} r_{N,0} \\ r_{N,1} \\ . \\ r_{N,N_1} \end{vmatrix}; ieF = LR \quad \text{---------------}\{2b\}$$

By column operations, (adding to each column the column on its left after multiplying by an arbitrary constant) V can be replaced by a matrix with polynomial elements as:-

$$\begin{vmatrix} P_0 & P_1(x_1) & . & P_N(x_1) \\ P_0 & P_1(x_2) & . & P_N(x_2) \\ . & . & . & . \\ P_0 & P_1(x_N) & . & P_N(x_N) \end{vmatrix}$$ Thus, (after a column and row additive operations on RHS and LHS in 2b)

$$\begin{vmatrix} \sum_0^N p_0(x_i)y_i \\ \sum_0^N p_1(x_i)y_i \\ . \\ \sum_0^N p_N(x_i)y_i \end{vmatrix} = \begin{vmatrix} \sum_{i=0}^N [P_0(x_i)]^2 & \sum_{i=0}^N P_0(x_i)P_1(x_i) & . & \sum_{i=0}^N P_0(x_i)P_N(x_i) \\ \sum_{i=0}^N P_0(x_i)P_1(x_i) & \sum_{i=0}^N [P_1(x_i)]^2 & . & \sum_{i=0}^N P_1(x_i)P_N(x_i) \\ . & . & . & . \\ \sum_{i=0}^N P_0(x_i)P_N(x_i) & \sum_{i=0}^N P_1(x_i)P_N(x_i) & . & \sum_{i=0}^N [P_N(x_i)]^2 \end{vmatrix} \begin{vmatrix} r_{N,0} \\ r_{N,1} \\ . \\ r_{N,N} \end{vmatrix}; \text{---}\{2c\}$$

Integer powers of x, x^i, or in general, a set of i^{th} degree (i=0 to n) polynomials formed from a linearly independent 'simple' function set of x can form the basis for such an approximation. Thus irrespective of whether discrete data is from a function (noise free) or from a measurement (noise may be present), it can have a polynomial approximation.

Optimum polynomial degree representation

Analytically, the interpolations described above ensure that the generated N degree polynomial coincides with arbitrary spaced N+1 data points. However, the lack of constraints at regions between the points, the loss of accuracy (due to overflow/ underflow, rounding off etc.) in computations, and other factors like noncompliant boundary conditions and discontinuous data, leads to fluctuations referred to as Runge & Gauss [10a-b] phenomena, close to the boundaries of x or at the discontinuities of y, respectively. Also, such a fit may be just providing a sketching through the data points. In reality, a meaningful fit may require only a lower polynomial of degree, n, (n<<(N+1)), with norms for ignoring out-liners (due to error / noise or approximation), sometimes with a transformed x-range. Of course, then (n<<(N+1)) the polynomial may pass through only few or even none of the points, so that one needs to look for a polynomial that passes through points 'closest' to the data. There are several approaches, [7] but examination of the least square method that employ linear sum of polynomials of various degrees, i (i= 0 to n), as basis for an incremental improvement of the fit (approximation) needs attention in the context of efficient attainability of the above requirement.

Least square fit to data

Let $p_n = \sum_{i=0}^n r_{n,i}\, O_i$ where $O_i = x^i$ (p_n is a n^{th} degree polynomial) or a more general $O_i = \sum_{j=0}^i k_{i,j}\, x^j$. For a n^{th} degree polynomial approximation, p_n, the measure of the error with in the least square approximation is :-

$$E_n = \sum_{k=0}^N \left[y_k - \sum_{i=0}^N r \quad O_i(x_k) \right]^2$$

This will have a minimum value with respect to r_{nj} when

$\frac{\delta E_n}{\delta r_{n,j}} = 2\sum_{k=0}^N \left[y_k - \sum_{i=0}^N r_{n,i}\, O_i(x_k) \right] O_j$ =0. ; (j=0 to n)

Setting $\frac{\delta E}{\delta r_{nj}} = 0$ yields the normal equation $\begin{vmatrix} F_0 \\ F_1 \\ . \\ F_n \end{vmatrix} = \begin{vmatrix} G_{11} & G_{12} & . & G_{1n} \\ G_{21} & G_{22} & . & G_{2n} \\ . & . & . & . \\ G_{n1} & G_{n2} & . & G_{nn} \end{vmatrix} \begin{vmatrix} r_{n,0} \\ r_{n,1} \\ . \\ r_{n,n} \end{vmatrix}$ $ieF = GR$ where $F_j = \sum_{k=0}^N y_k O_j(x_k)$ and $G_{i,j} = \sum_{k=0}^N O_i(x_k) O_j(x_k)$ ---------------{3}

Thus $R=G^{-1} F$

The inversion of the Gram matrix, G, to get R (r_{nj}) is not needed (a computational advantage) if, the set {O} satisfy

$\sum_{k=0}^N O_i(x_k) O_j(x_k) = h_j \delta_{ij}$ ----------------{4}

This defines discrete orthogonality. This discrete sum and the integral $\int_l^u O_i\, O_j dx = h_i \delta_{i,i}$ are interchangeable if the function set O_j is singularity free in the x range [l,u], not only represent the data points, y_i at x_i , but also is a good interpolation between the points, and each member of the set and its derivative satisfy certain boundary conditions (section 2.2). Since with discrete orthogonality, the Gram matrix is diagonal:-

$r_{n,j} = \frac{F_j}{G_{j,j}}$. --------------------{5}

When x^i is used as the basis, (the non-orthogonal polynomial) case the matrix equation corresponding to the least square norm takes the form:-

$$\begin{vmatrix} \sum_0^N y_i \\ \sum_0^N x_i y_i \\ . \\ \sum_0^N {x_i}^n y_i \end{vmatrix} = \begin{vmatrix} N+1 & \sum_0^N x_i & . & \sum_0^N x_i^n \\ \sum_0^N x_i & \sum_0^N {x_i}^2 & . & \sum_0^N x_i^{n+1} \\ . & . & . & . \\ \sum_0^N {x_i}^n & \sum_0^N x_i^{n+1} & . & \sum_0^N x_i^{2n} \end{vmatrix} \begin{vmatrix} r_{N,0} \\ r_{N,1} \\ . \\ r_{N,n} \end{vmatrix}; ieF = LR \text{------}\{6\}$$

Thus, $r_{N,j} = L^{-1} \sum_{k=0}^{N} x_i{}^j y_i(x_k)$ (in the general form, the elements of L are integrals). This requires matrix inversion and re-computation for any change in n. However, in the case of orthogonal polynomials, the $r_{n,j}$ are independent of n and hence no re-computation of $r_{n,j}$ is required as the trial degree is incremented to n+1 (<N); another computational advantage. Note that for any member of the set $\{p_N\}$, p_n ($0 \leq n \leq N$), the discrete orthogonality is defined by a summation over the roots of p_N. Also each p_n is the best n^{th} degree polynomial fit in the least-square sense and will be a better approximation as compared to any lower degree polynomials.

Since $\begin{vmatrix} N+1 & \sum_0^N x_i & . & \sum_0^N x_i^n \\ \sum_0^N x_i & \sum_0^N x_i{}^2 & . & \sum_0^N x_i^{n+1} \\ . & . & . & . \\ \sum_0^N x_i{}^n & \sum_0^N x_i^{n+1} & . & \sum_0^N x_i^{2n} \end{vmatrix} = \begin{vmatrix} 1 & 1 & . & 1 \\ x_0 & x_1 & . & x_N \\ . & . & . & . \\ x_0^n & x_1^n & . & x_N^n \end{vmatrix} \begin{vmatrix} 1 & x_0 & . & x_0^n \\ 1 & x_1 & . & x_1^n \\ . & . & . & . \\ 1 & x_N & . & x_N^n \end{vmatrix}$ and $\begin{vmatrix} 1 & 1 & . & 1 \\ x_0 & x_1 & . & x_N \\ . & . & . & . \\ x_0^n & x_1^n & . & x_N^n \end{vmatrix}$ can be converted to $\begin{vmatrix} p_0(x_0) & p_0(x_1) & . & p_0(x_N) \\ p_1(x_0) & p_1 x_1 & . & p_1 x_N \\ . & . & . & . \\ p_n(x_0) & p_n(x_1) & . & p_n(x_N) \end{vmatrix}$ one can rewrite the expression for least square as :-

$$\begin{vmatrix} \sum_0^N p_0(x_i) y_i \\ \sum_0^N p_1(x_i) y_i \\ . \\ \sum_0^N p_n(x_i) f_i \end{vmatrix} = \begin{vmatrix} \sum_{i=0}^{N-1} [P_0(x_i)]^2 & \sum_{i=0}^{N-1} P_0(x_i) P_1(x_i) & . & \sum_{i=0}^{N-1} P_0(x_i) P_n(x_i) \\ \sum_{i=0}^{N-1} P_0(x_i) P_1(x_i) & \sum_{i=0}^{N-1} [P_1(x_i)]^2 & . & \sum_{i=0}^{N-1} P_1(x_i) P_n(x_i) \\ . & . & . & . \\ \sum_{i=0}^{N} P_0(x_i) P_n(x_i) & \sum_{i=0}^{N} P_1(x_i) P_n(x_i) & . & \sum_{i=0}^{N} [P_n(x_i)]^2 \end{vmatrix} \begin{vmatrix} r_{n,0} \\ r_{n,1} \\ . \\ r_{n,n} \end{vmatrix} \text{----}\{7\}$$

Thus, if $\sum_{k=0}^{N} P_i(x_k) P_j(x_k) = h_j \delta_{ij}$, it is equivalent to the use of an orthogonal set ($\{p_j\} 0 \leq \mathbf{j} > N$, with a discrete orthogonality) with the roots of P_N as the grid and automatically ensure a least square fit for any n<N. The discrete orthogonality instead of the orthogonality integral, leads to efficient numerical computations.

Construction of orthogonal polynomials from discrete data.

From the procedure to generate polynomials from discrete data discussed above, it is clear that it is defined only at discrete (grid) points of the independent variable and a set of finite degree polynomials are needed for approximations of various degree. To generate such a finite orthogonal set, two approaches are used; use an x-grid that are the N roots of a 'suitable' orthogonal polynomial, p_N, or construct an orthogonal polynomial that have N equal spaced (or in general q-spaced [11a-c]; not considered here) roots. These are respectively known as continuous and discrete cases. In the continuous case, the grid is non-uniform and the needed data is generated from the complex function that is being approximated to a member of the orthogonal polynomial set. In the discrete case, the equal spaced data (special q-spaced) points are available at the grid points by construction. Apart from the restriction on grid spacing, the orthogonal set can be identified as the solution of certain class differential/difference equations [1,12] that are reducible to a conjugate form. Eigen value equations $\left[\frac{d^2y}{dz^2} - \lambda_n y = 0\right]$ with trigonometric functions as eigen vectors and positive real discrete values as eigen values are special cases of such equations in conjugate form. The link between orthogonal sets and differential equations reveal the boundary conditions required and several analytical relations. This also increases the available orthogonal sets and facilitates in getting approximate solutions to several related differential equations.

III. Orthogonal Polynomials Of Hahn's Class

Several computational Physics and mathematics problems are formulated as a difference or differential equations [13a-b] that are limiting cases of q-difference equations expressed in terms of Hahn's operator [14a-b]. The Hahn's operator $\delta y_{(x)} = \frac{y_{(qx+r)} - y_{(x)}}{(q-1)x+r}$ (q and r are real numbers with $q \neq 1$ $r \neq 0$ and $x \neq \frac{r}{1-q}$) satisfy a degree conserving (all terms of same degree) q-difference equation of the form:-

$\sigma \delta^2 y\left(\frac{1}{q}(x-r)\right) + \tau_{n0} \delta y\left(\frac{1}{q}(x-r)\right) = \lambda_n y$ with σ and τ_{n0} polynomials of at most two and one degree respectively [$\sigma = \sigma_2 x^2 + \sigma_1 x + \sigma_0; \tau_{n0} = \tau_1 x + \tau_0$]. Since δ can be the normal differential operator (for $\lim_{q\to 1} \delta\, y$) or the difference operator (Δ or ∇ when $q \to 1$ and $r = \pm 1$), the above equation reduces to two sub classes, referred to respectively as continuous and discrete cases.

$\sigma y^{!!} + \tau_{n0} y^{!} + \lambda_{n0} y = 0$ -------------------------{8}

$\sigma(\Delta \nabla y) + \tau_{n0} \Delta y + \lambda_{n0} y = 0$ -------------------------{9}

Here only these two sub-classes will be discussed and not the general q difference case [11a-c].

In general equation of the form {8} can be obtained via a transformations of the more general equation $y^{!!} + \frac{\tau_{n0}}{\sigma} y^{!} + \frac{\Sigma}{\sigma^2} y = 0$ where Σ and σ have degree $\leq \mathbf{2}$ and τ is of degree one [13b] and encompasses large number of cases of scientific interest.

Properties of differential equations

The equation {8} has, in addition to a singularity at ∞, singularities at the roots, s, of σ or τ_{n0} (s given by $\sigma_2 s^2 + \sigma_1 s + \sigma_0 = 0$ or if $\sigma_2 = \sigma_1 = 0 \sigma_0 \neq 0$, s=$\frac{-\tau_0}{\tau_0}$). Thus a power series solution (Frobenius method), of the form $y_n = \sum_{i=0}^{n} k_{n,i} \frac{(x-s)^{e+i}}{i!}$ (e, the exponent corresponding to the singularity at s) should exist. This, for the positive real polynomials, y_n leads to a recursion relation for $k_{n,i}$ (for 0≥n≤∞.; with $k_{n,n}$=1) [11a] as:-

$(n-i)[\sigma_2(n+i-1)+\tau_1]k_{n,i} - [(2\sigma_2 s+\sigma_1)i+\tau_1 s+\tau_0]k_{n,i+1} - (\sigma_2 s^2+\sigma_1 s+\sigma_0)k_{n,i+2} = 0$ ---------- {10}

This will reduce to a two term recursion if $(\sigma_2 s^2 + \sigma_1 s + \sigma_0)$ or $\tau_1 x + \tau_0$ equals zero with the region between the roots of σ (or τ_{n0} as the case be) having singularity free solutions. It also fixes zero as one of the two possible values of the exponent e. With such a monic polynomial solution for {8}, λ_{n0} can be obtained as $-n[\sigma_2(n-1)+\tau_1]$ (by equating the coefficients of the zeroth power of x with such a trial solution). These are in fact the conditions for the existence of polynomial solutions to {8}. Such monic polynomial solution of {8} satisfies the condition on the ratio of two adjacent $k_{n,i}$ required for the existence of a hypergeometric representation [3a-c,11a]. Thus it is obvious that a set of polynomials of definite degree n, y_n, $\left(\left[y_n = \sum_{i=0}^{n} k_{n,i} \frac{(x-s)^i}{i!}\right]; k_{n,n} = 1\right)$ are the particular solutions of {8} and with an appropriate x range, will have a hypergeometric representation. Such particular solutions, p_n (i.e. y_n=$p_n = \sum_{i=0}^{n} k_{n,i} \frac{(x-s)^i}{i!}$; n=0,1...∞) as a basis set, the converging series $\sum_{i=0}^{\infty} r_{n,i}\, p_i$ (=y) is one of the general solutions of {8}. As it turnout, {8} is recast-able to a conjugate form and among many other properties, these solutions form an orthogonal set.

Self conjugate form, weighted solution and orthogonality

In {8}, if $\sigma^! = \tau_{n0}$, it takes the self-conjugate (eigen value equation) form [12]. If the condition $\sigma^! = \tau_{n0}$ is not met, multiplication of y by a function w that satisfy the condition $w\sigma^! = w\tau_{n0}$ will recasts {8} into a conjugate form [12].

Thus $\sigma y^{!!} + \tau_{n0} y^! + \lambda_{n0} y = 0$ takes the form

$$\frac{d}{dz}[w\sigma y^!] + w\lambda_{n0} y = 0$$

Provided $(w\sigma)^! = w\tau_{n0}$

or $w^! = w\frac{(\tau_{n0}-\sigma^!)}{\sigma}$.------------------------{11a}

This implies that:- $w = \frac{1}{\sigma} exp \int_l^u \left(\frac{\tau_{n0}}{\sigma}\right) dx$ -------------------{11b}

The limits of integration, l & u, needs to ensure that w is positive and finite. Since {11b} implies that σ should not be zero within the limits of integration, l & u needs to be the roots of σ, if σ is a second degree polynomial. In other cases also, the limits are decided by the location of the singularities of {8}.

If p_n and p_m are two solutions of {8} with eigen values λ_u and λ_v, then

$\int_l^u p_n\, p_m w dx = h_n \delta_{nn}$ -------------{12}

provided a suitable boundary condition sets $[w\sigma(p_n^! p_m - p_n p_m^!)]_l^u$ to zero [12].

Thus, P_n, w, $\lambda_{n,0}$, boundary conditions and restricting the range of x to singularity free region define a general orthogonal set that satisfy $\int P_n w P_m dx = h_n \delta_{nm}$. The generated set will be unique with an additional condition that the leading coefficient of P_n is positive. It may be noted that such solutions require specific x range ([-1,1], [0,1], [0,∞],[-∞ ∞] etc.) in the hypergeometric representation. Thus data will need a linear transformation to have such compatible range and the use of such standard range has an advantage in numerical computations also.

Orthogonality implies that like y, any function F, (including x^m, P_m etc) may be approximated by an orthogonal polynomial set as:- $F_n = \sum_{i=0}^{m} r_{n,i}\, P_i$ (m=n if F is a polynomial of degree n, ∞ otherwise) because $r_{n,i}$ can always be obtained as:-

$r_{n,i} = \left[\int F_n\, p_j w\right] dx \div \left[\int p_j\, p_j w dx\right]$ ------------------------{13}

Similarly, $\int x^n P_m w dx = [h_n \div k_{n,n}]\delta_{nm}$ $\int x p_{n-1} P_n w dx = [k_{n-1,n-1} h_n] \div k_{n,n}$ with $p_n = \sum_{i=0}^{n} k_{n,i}\, x^i$

Discrete sums and Orthogonality of finite polynomial sets

As discussed earlier, for an integer N, a finite polynomial set can be generated that satisfy a discrete orthogonality. If p_N, p_n and p_m (m,n≤ N) belongs to such a set, then the orthogonality integral, because then p_n is the best fit to N points in the least square sense (see section1.4), takes the (the superscript i implies the value of p_n at the i^{th} root of p_N) form:-

$\sum_{i=1}^{N} w\, p_n^i p_m^i = h_n \delta_{nm}$ ---------------------------{14a}

This expression also implies the existence of another discrete sum as [15] :-

$\sum_{k=1}^{N} \frac{wp_k^i p_k^j}{h_k} = h_i \delta_{ij}$ -----------------------------{14b}

The equations {14a &b} are the discrete form of the orthogonality integral {12}

The Christoffel-Darboux identity and Gauss quadrature also are stated using discrete sums (section2.9 and [17,17]).

Recursion relation

A consequence of orthogonality is that the RHS of the expansion $xP_n = \sum_{i=0}^{n+1} r_{n+1,i} P_i$ takes the form $a_{n+1}xP_n - b_{n+1}P_n - c_{n-1}P_{n-1}$ (because xP_n is a polynomials of degree n+1) with:-

$a_{n+1} = \frac{\int P_{n+1}P_{n+1}wdx}{\int (xP_n)P_{n+1}wdx}$; $b_{n+1} = a_{n+1}\frac{\int P_n xP_n wdx}{\int P_nP_nwdx}$; $c_{n-1} = \frac{[a_{n+1}\int P_n(xP_{n-1})wdx]}{[\int P_{n-1}P_{n-1}wdx]} = \frac{a_{n+1}}{a_n}$ ----{15}.

Thus $xP_n = \frac{P_{n+1}}{a_{n+1}} + \frac{b_{n+1}}{a_{n+1}}P_n + \frac{P_{n-1}}{a_n}$ or $P_{n+1} = (a_{n+1}x - b_{n+1})P_n - c_{n-1}P_{n-1}$ ------------------{16}

(the -ve signs used for keeping b_n & c_n positive in subsequent relations)

For finite set of orthogonal polynomials, the summation over N replaces the integrals.

The above recursion can be initiated with $P_0 = h_0; p_1 = (a_0x - b_0)P_0$ and is the most efficient means of generating an orthogonal polynomial set.

Recursion and other relations for the polynomial.

Useful relations for the general polynomial expansion coefficients $k_{n,j}$ $\left(p_n = \sum_{i=0}^{n} k_{n,i}\, x^i\right)$ can be obtained by equating the coefficients of x^l in the recursion {16}:-

$$\sum_{i=0}^{n+1} k_{j+1,i}\, x^{i+1} = a_{n+1}\sum_{i=0}^{n} k_{n,i}\, x^{i+1} - b_{n+1}\sum_{i=0}^{j} k_{n,i}\, x^i - c_{n-1}\sum_{i=0}^{j-1} k_{n-1,i}\, x^i.$$

These are :-

$$k_{n+1,n+1} = a_{n+1}k_{n,n}; l = n+1$$

$$k_{n+1,n} = a_{n+1}k_{n,n-1}. -b_{n+1}k_{n,n}; l = n$$

$$k_{n+1,l} = a_{n+1}k_{n,l-1}. -b_{n+1}k_{n,l} - c_{n-1}k_{n-1,l}; l = n-1, n-2, \ldots\ldots 2$$ ------------------(17a)

Or in general :-

$$k_{0,0} = p_0$$

$$k_{i+1,i+1} = a_{i+1}k_{i,i}; i = 0,1, ---N$$

$$k_{i+1,i} = a_{i+1}k_{i,i-1}. -b_{i,1}k_{i,i}; i = 1,2, ---N$$

$$k_{i,j} = a_ik_{i-1,j-1}. -b_ik_{i-1,j} - c_{i-2}k_{i-2,j}; i = 2,3 ----- N \quad j \leq i$$ -------------{17b}

It may be noted that the indexing and sign of a, b & c are in accordance with that used in the recursion relation {16} and if it is different, as in some literature, those used here should be accordingly readjusted.

Several relations between the recursion coefficients (a_n, b_n & c_n) and between the recursion and the polynomial coefficients can be obtained from equations {15} & {17a-b}.

$a_{n+1} = \frac{k_{n+1,n+1}}{k_{n,n}} \Leftrightarrow \int (xP_n)P_{n+1}wdx = h_{n+1}\frac{k_{n,n}}{k_{n+1,n+1}}$

Thus using the two alternate expressions of a_n

$$h_{n+1} = \frac{k_{n+1,n+1}^2}{k_{n,n}^2} = a_{n+1}^2$$

$$b_{n+1} = a_{n+1}\left[\frac{k_{n,n-1}}{k_{n,n}} - \frac{k_{n+1,n}}{k_{n+1,n+1}}\right]$$

$$c_{n-1} = \frac{[a_{n+1}\int P_n(xP_{n-1})wdx]}{[\int P_{n-1}P_{n-1}wdx]} = \frac{h_n}{h_{n-1}}\frac{k_{n+1,n+1}k_{n-1,n-1}}{k_{n,n}^2} = \frac{h_n}{h_{n-1}}\frac{a_{n+1}}{a_n}$$

Also, $a_{n+1} = a_na_{n-1}\ldots\ldots a_0$; $a_0 = \frac{k_{1,1}}{k_{0,0}}$; $c_{n-1} = c_{n-2}c_{n-1}\ldots\ldots c_0$; $c_0 = \frac{k_{2,2}k_{0,0}}{k_{1,1}^2}\frac{h_1}{h_0}$

The above listed relations imply that, b can be zero and a & c are not independent variables.

In general, the recursion formula for the generation of the polynomial set takes the form:-

$p_{n+1} = (a_{n+1}x - b_{n+1})p_n - \frac{h_n}{h_{n-1}}\frac{a_{n+1}}{a_n}p_{n-1}$ --------------{18}

The recursion coefficients a_n, b_n & c_n can also be obtained in terms of the coefficients of the polynomials σ and τ_{n0} [18a-b] and is given in section2.8 for monic case[11a].

Orthogonality of Derivatives

An important property of the solution of {8} is that all the derivatives of y are also solutions of it, as its structure is conserved under differentiation [15]. For the m^{th} derivative (m≤n-1) it takes the form:-

$\sigma v^{!!}{}_k + \tau_{nk} v^!_k = \lambda_{nk} v_k; o > k \le n-1$ (with v_k as the k^{th} derivative of y)
$\tau_{nk} = \tau_{n0} + \sigma^! k; \lambda_{nk} = \lambda_{n0} + \tau^!_{nk} + \sigma^{!!}(k-1)k/2$
Also $w_k = w\sigma^k$
Thus the orthogonality integral of the k[th] derivative
$\int P^k_m P^k_n w_k dz = h^k_{n,n}\delta_{nm}$ can be evaluated from the relation $h^{i+1}_{nn} = \lambda \ \ h^i{}_{nn}$ starting with $h^0_{nn} = \int p^0_n p^0_n w dz$

Recursion relation of k[th] derivative of P_n can be obtained by differentiating {12} as:-

$$P^k_{n+1} = (a_{n+1}z - b_{n+1})P^k_n - c_{n-1}P^k_{n-1} + a_{n+1}kP^{k-1}_n$$

(with no negative subscript/superscript allowed in the RHS.). These relations lead to Rodrigues formula as [15]:-
$p_n \propto \frac{1}{w}\frac{d^n(w\sigma^n)}{dx^n}$

The I[st] (and all higher) derivative of P_n can also be generated [19] by$\sigma P^!_n = \alpha_n P_{n+1} + \beta_n P_n + \gamma_n P_{n-1}$ with α_n, β_n and γ_n computable from the coefficients of σ and τ [19].

Some useful summations

For the evaluation of finite sums of the form $F = \sum_{i=0}^{n} r_{n,i}\, p_i$ Clenshaw Algorithm [20a-b] is often used. As per this algorithm, by virtue of the recursion relation for P_i, as $P_i = (a_i x - b_i)P_{i-1} - c_{i-1}P_{i-2}$, another recursion $B_i = A_i + (a_{i+1}x - b_{i+1})B_{i+1} - c_{i+1}B_{i+2}$ (with B_i=0 for i>n) exists. The recursion is initiated (from i=n) with $A_n=B_n$ to get B_0 and then, $F=P_0B_0$..

The Horner recursive algorithm computes polynomials (including orthogonal) with better precision [21] by avoiding computation of high powers of x.

Extension to more general case.

In the more general case when the independent variable is a function of x, (f), {8} takes the form.

$$\sigma^! f^! y^{!!} + [\tau^!{}_{n0} f^! f^! - \sigma^! f^{!!}]y^! + \lambda_n f^{!!} f^! f^! y = 0$$

Following the procedure as in the case of {8} one can get the weight function w as :-
$\frac{1}{\sigma^! f^!} exp \int_{l_n}^{u_n} \left(\frac{\tau^!{}_{n0} f^! f^! - \sigma^! f^{!!}}{\sigma^! f^!}\right) df$

For the case when f is obtained as a result of a linear transformation of x ($f=l_1x+l_0$, $f^! = l_1$ and $f^{!!} = 0$), the weight functions will differ by a factor l_1 and the degree of σ and τ_{n0} will remain unaltered. This implies that all results discussed above are applicable in this transformed case also with changed limits of integration and the transformed σ, τ_{n0} and w. As an example, a linear transformations ($z_i = \epsilon_1 x_i + \epsilon_0$ or the reverse $x_i = \frac{1}{\epsilon_1}[z_i - \epsilon_0]$; $\epsilon_1 = 2/[x_N - x_1]$; $\epsilon_0 = -[x_N + x_1]/[x_N - x_1]$),will change the limits of integration between $[x_1,x_N]$ and [-1,1], but will leave a_n, b_n & c_n unaltered because $\left[\int_{x_1}^{x_n} f(x) w dx\right] = \left[\int_{-1}^{1} f(z)\frac{w}{\epsilon_1} dz\right]$. Since the above transformation implies, $x_i = \eta_1 z_i + \eta_0$ [with $\eta_1 = 0.5(x_N - x_1)$;$\eta_0 = 0.5(x_N + x_1)$], the recursion changes as
$P_{n+1} = [\eta_1 a_{n+1} z - (b_{n+1} - \eta_0 a_{n+1})]P_n - c_{n-1}P_{n-1}$---------------------{19}

The reverse transformation is $x_i = \eta_1 z_i + \eta_0$; $\eta_1 = 0.5(x_N - x_1)$;$\eta_0 = 0.5(x_N + x_1)$ and may also be used to obtain the grid for other intervals ([0,1], [0,N] and [0, ∞]). Thus orthogonal polynomials can be computed in any convenient range (related through a linear transformation), but as a standardization, results are often presented in terms of range compatible with hypergeometric representation.

It is possible to assign P_0 any value that is independent of x, to get compact relations, or numerical stability in computations.

Two forms of orthogonal polynomials that are often used are the Monomial and Orthonormal and respectively correspond to cases where all k_n or h_n are set to unity.

1) Monomial

$P_{n+1} = (z - b_{n+1})P_n - \frac{h_n}{h_{n-1}}P_{n-1}$ $\quad p_{n+1} = [z - (k_{n,n-1} - k_{n+1,n})]p_n - \frac{h_n}{h_{n-1}}p_{n-1}$ -------{20}

with p_0=1

The recursive relation obtainable from the orthogonality for the monic case in terms of the coefficients of σ and τ_{n0} is [11a]:-

$y_{n+1} = (z - b_{n+1})y_n - c_{n-1}y_{n-1}$ with $b_{n+1} = \frac{n\sigma_1[\sigma_2(n-1)+\tau_1] - \tau_0(\sigma_2 - \tau_1/2)}{2[\sigma_2(n-1)+\tau_1/2][\sigma_2 n + \tau_1/2]}$

$$c_{n-1} = \frac{n[\sigma_2(n-2)+\tau_1][\sigma_1(n-1)+\tau_0\sigma_2\sigma_1(n-1)+\sigma_1\tau_1-\sigma_2\tau_0] - 4\sigma_0[\sigma_2(n-1)+\tau_1/2]^2}{4[\sigma_2(2n-3)+\tau_1][\sigma_2(n-1)+\tau_1/2]^2[\sigma_2(2n-1)+\tau_0]}$$

or ($b_{n+1} = \frac{-\tau_0}{\tau_1}$ $\quad c_n = \frac{-n}{\tau_1}$ when $\sigma_2 = \sigma_1 = 0; \sigma_0 \neq 0$).

2) Orthonormal

$p_{n+1} = (a_{n+1}z - b_{n+1})p_n - \frac{a_{n+1}}{a_n}p_{n-1}$ or $0 = zp_n - \left[\frac{p_{n+1}}{a_{n+1}} + \left(\frac{b_{n+1}}{a_{n+1}}\right)p_n - \frac{p_{n-1}}{a_n}\right]$----------{21}

It may be noted that in general $p_0 = k_{0,0}$ and $h_0 = \int p_0\, p_0 wdz = \int k_{0,0}^2\, wdz$. If h_0 is not one, it possible to scale P_0 and get a scaled $\bar{p_0}$ ($\bar{p_0} = \frac{P_0}{\sqrt{h_0}}$) and generate a recursion relation starting with $\bar{p_0} = \frac{1}{\sqrt{h_0}} : \bar{p_1} = \frac{p_1}{\sqrt{h_0}}$. and

$\bar{p}_{n+1} = (a_{n+1}z - b_{n+1})\bar{p}_n - \frac{a_{n+1}}{a_{n-1}}\bar{p}_n$ -----------{22}

For conversion between monomial and orthonormal forms the following relations can be employed. If $M_{n+1} = (z - b_{n+1})M_n - \gamma_n M_{n-1}$ is a monomial set, then the corresponding orthonormal set is:-

$O_{n+1} = \left[(z - b_{n+1})O_n - \sqrt{\gamma_n}O_{n-1}\right]/\sqrt{\gamma_{n+1}}$ ------------------{23}

The equations for the orthonormal polynomials p_1 to p_{n+1} generated by the recursion relation, after rearranging it as $\frac{p_{n+1}}{a_{n+1}} = \left[\frac{p_{n-1}}{a_n} + \left(\frac{b_{n+1}}{a_{n+1}}\right)p_n\right] - xp_n$ and written in a matrix form is:-

$$\begin{vmatrix} 0 \\ 0 \\ . \\ . \\ 0 \\ P_{n+1}/a_{n+1} \end{vmatrix} = \left(\begin{vmatrix} b_1/a_1 & 1/a_1 & 0 & . & 0 & 0 \\ 1/a_1 & b_2/a_2 & 1/a_2 & . & 0 & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & 0 & . & b_n/a_n & 1/a_n \\ 0 & 0 & 0 & . & 1/a_n & b_{n+1}/a_{n+1} \end{vmatrix} - \begin{vmatrix} z & 0 & 0 & . & 0 & 0 \\ 0 & z & 0 & . & 0 & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & 0 & . & z & 0 \\ 0 & 0 & 0 & . & 0 & z \end{vmatrix} \right) \begin{vmatrix} p_0 \\ p_1 \\ . \\ . \\ p_{n-1} \\ p_n \end{vmatrix}$$

If we consider that z takes only the values that corresponds to the roots of p_{n+1}, z_i, then for each such z_i the LHS in the above equation is zero (because at $z=z_i$ $p_{n+1}=0$). Thus on the RHS requires that:-

$$\left(\begin{vmatrix} z_i & 0 & 0 & . & 0 & 0 \\ 0 & z_i & 0 & . & 0 & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & 0 & . & z_i & 0 \\ 0 & 0 & 0 & . & 0 & z_i \end{vmatrix} - \begin{vmatrix} b_1/a_1 & 1/a_1 & 0 & . & 0 & 0 \\ 1/a_1 & b_2/a_2 & 1/a_2 & . & 0 & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & 0 & . & b_n/a_n & 1/a_n \\ 0 & 0 & 0 & . & 1/a_n & b_{n+1}/a_{n+1} \end{vmatrix} \right) = 0 \quad \text{----------\{24\}}$$

The above can be identified as the eigen value equation of the matrix J, known as Jacobi Matrix.

$$J = \begin{vmatrix} b_1/a_1 & 1/a_1 & 0 & . & 0 & 0 \\ 1/a_1 & b_2/a_2 & 1/a_2 & . & 0 & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & 0 & . & b_n/a_n & 1/a_n \\ 0 & 0 & 0 & . & 1/a_n & b_{n+1}/a_{n+1} \end{vmatrix} \quad \text{-----------------------------\{25\}}$$

These eigenvalues (roots/grid points) and the corresponding vectors play an important role in the quadrature methods that employ orthogonal polynomials.

Christoffel-Darboux identity.

A relation involving discrete sums, known as Christoffel-Darboux identity, exists for two polynomial sets $\{p_N(z_1)\}$ and $\{p_N(z_2)\}$. The two variables, z_1 and z_2 are related via a linear transformation and thus have the same recursion coefficients. For the general (non-monomial and non-orthonormal) sets [16,22,23] it has the following form.

$\sum_{i=0}^{N-1} \frac{p_i(z_1)p_i(z_2)}{h_i} = \frac{1}{a_N h_{N-1}} \frac{[p_{N-1}(z_2)p_N(z_1) - p_{N-1}(z_1)p_N(z_2)]}{(z_1 - z_2)}$ ------------{ 26a}

It takes a simpler form for the orthonormal case as:-

$\sum_{i=0}^{N-1} O_i(z_1)O_i(z_2) = \frac{1}{a_N} \frac{[O_{N-1}(z_2)O_N(z_1) - O_{N-1}(z_1)O_N(z_2)]}{(z_1 - z_2)}$ ----------------{ 26b}

In particular when $z_1 = z_2 = z$

$$\sum_{i=0}^{N-1} O_i(z)O_i(z) = \frac{1}{a_N}[O_{N-1}(z)O_N^!(z) - O^!{}_{N-1}(z)O_N(z)]$$

If the discrete values of z corresponds to the N roots of O_N (i.e O_N=0), z_j, this yields [17(Eq35, chapter 2)] :-

$\sum_{i=0}^{N-1} O_i(z_j)O_i(z_j) = \frac{1}{a_N} O_{N-1}(z_j)O_N^!(z_j)$ --------------------------------{27a}

The corresponding equation for the monomial will be:-

$\sum_{i=0}^{N-1} \frac{p_i(z_j)p_i(z_j)}{h_i} = \frac{1}{h_{N-1}} p_{N-1}(z_j)p_N^!(z_j)$ --------------------------------{27b}

This expression leads to an efficient computation of weight in quadrature methods.

Generating new weight functions.

If w is a weight function that yield an orthogonal set $\{p_n\}$ then $\Omega = \varphi\, w$, also satisfies the requirements of the weight function if φ is a non-negative polynomial of degree L within the limits of integration [22]. The new orthogonal set that corresponds to Ω is $\psi = \frac{1}{\varphi} D_n$ where D_n is :-

$$D_n = \begin{vmatrix} p_n(x) & p_{n+1}(x) & . & p_{n+L}(x) \\ p_n(r_1) & p_{n+1}(r_1) & . & p_{n+L}(r_1) \\ . & . & . & . \\ p_n(r_L) & p_{n+1}(r_L) & . & p_{n+L}(r_L) \end{vmatrix}$$ **(**r_i are the L roots of φ **)**

Additional ways of generating orthogonal sets

In general, if a function set, Ω_i (i=1….n) and it and its derivatives, $\Omega_i^!$, exists, then the n[th] degree polynomial p_n given by Rodrigues formula , $p_n = \frac{1}{w}\frac{d^n \Omega_n}{dz^n}$, form an orthogonal polynomial set [22, page 59] for a weight function, w, provided Ω_i and derivatives vanish at the boundaries. The boundary conditions on Ω_j, $\Omega_i^!$ and $\frac{d^{n+1}p_n}{dz^n+1} = 0.$ in fact identifies the set Ω_n. When $\Omega_n = w\sigma^n$, P_n satisfy differential equation {8}.

For a given w, orthogonal polynomial set, can be generated via the moments matrix, constructed from the finite integrals $m_i = \left[\int_l^u w\, z^i dx\right]$. Let A & B be:-

$$A=\begin{bmatrix} 1 & z & . & z^{n-1} & z^n \\ m_1 & m_2 & . & m_n & m_{n+1} \\ . & . & . & . & . \\ m_{n-1} & m_n & . & m_{2n-2} & m_{2n-1} \\ m_n & m_{n+1} & . & m_{2n-1} & m_{2n} \end{bmatrix} \quad B=\begin{bmatrix} m_0 & m_1 & . & m_{n-1} & m_n \\ m_1 & m_2 & . & m_n & m_{n+1} \\ . & . & . & . & . \\ m_{n-1} & m_n & . & m_{2n-2} & m_{2n-1} \\ m_n & m_{n+1} & . & m_{2n-1} & m_{2n} \end{bmatrix}.$$

Then P_n=A/B will form an orthogonal set because $\int_l^u p_n\, p_m w dz = \delta_{nn}$. This approach can be employed to generate a set from any suitable w and need not always satisfy differential equation of the form {8}.

Starting with any suitable linearly independent set $\{v_j\}$ of functions treated as vectors, one can generate an orthogonal polynomial set $\{x_i\}$ with the same span as $\{v_j\}$ via Gram-Schmidt scheme [24] as:-

$$x_i = v_i - \sum_{j=1}^{i-1} \frac{\langle v_i | x_j \rangle}{\langle x_j | x_j \rangle} x_j$$

These general approaches may need the computation of recursion coefficients via integrals.

Classical orthogonal polynomials

Classical polynomials (so named because they were known for a long time as solutions of some physics problems that have differential equations of the form {8}), have an associated infinitely differentiable function Ω_i with an auxiliary condition $\Omega_n = w\sigma^n$ [22,section 2.7]. They give standard expressions (avoiding integration) for a_n, b_n and c_n and weight function, and are widely used because of the efficient computational possibility. As mentioned earlier, for variables related by linear transformation, the orthogonal set is not new except for a modified recursion relation and weight. From the discussion up to now it is clear that the solutions of {8} with restrictions on the degree and coefficients of σ and τ meet all these requirements. They can also be made to meet the independent variable range compatible with hypergeometric representation. Thus one can look for the hypergeometric form of particular solutions of {8}. The requirement of a positive-Definite weight and the possible solutions of σ=0 or τ=0 give rise to three unique monic Orthogonal Polynomial Solutions [11a].

case 1 degree of σ zero implying $\sigma_2 = \sigma_1 = 0, \sigma_0 = 1$, $\tau_1<0$ and x range $[-\infty, \infty]$

The differential equation has the form:- $y^{!!} + (\tau_1 x + \tau_0)y^! = \tau_1 n$

The weight function obtained from the relation $w^! = w\frac{(\tau_{n0} - \sigma^!)}{\sigma}$ is:- $w = exp\left(\frac{x^2}{2}\tau_1 + \tau_0\right)$;

The recursion is:- $y_{n+1} = (x - b_{n+1})y_n - c_{n-1}y_{n-1}$ with $b_{n+1} = \frac{-\tau_0}{\tau_1}\, c_n - 1 = \frac{-n}{\tau_1}$ $\tau_1<0$

The polynomial expansion takes the form:-

$y_n = \sum_{i=0}^{\infty} r_{n,i}\left(x + \frac{\tau_0}{\tau_1}\right)^i$ with $r_{n,n-2i+1} = 0; r_{n,n-2i} = \frac{n!}{2{\tau_1}^2 i!}$

The hypergeometric form is:- $y_n = \left(x - \frac{\tau_0}{\tau_1}\right)^n {}_2F_0\left(\frac{-n}{2}, \frac{-(n-1)}{2}n; -; \frac{2}{\tau_1\left(x - \frac{\tau_0}{\tau_1}\right)^2}\right)$

Orthogonality integral is:-$\int_{-\infty}^{\infty} w\, y_n y_m dx = \sqrt{\frac{2\pi}{\tau_1}\frac{n!}{(-\tau_1)^n}}\, exp\left(\frac{\tau_0^2}{2\tau_1}\right)\delta_{n,n}$

This corresponds to Hermite polynomials

Case 2 degree of σ is one implying σ_2 =0 and σ_1=1 and x range $[-\sigma_0, \infty]$

If $\tau_1\sigma_0 < \tau_0$ and with a parameter α ($\tau_0 + \sigma_0\tau_1 - 1$ which implies that α+1>0), the weight w ($w = (x+\sigma_0)^\alpha exp(\tau_1 x)$),is positive. The differential equation has the form :- $(x-l)y_n^{!!} + (\tau_1 x - \tau_1\sigma_0 + \alpha + 1)y_n^{!} = \tau_1 n y_n$ with singularity at -σ_0(=l).

The recursion is $y_{n+1} = (x - b_{n+1})y_n - c_{n-1}y_{n-1}$ with $b_{n+1} = \frac{-(2n+\tau_0)}{\tau_1}$ $\quad c_{n-1} = \frac{n(n-1)+\tau_0-\tau_1\sigma_0}{\tau_1\tau_1}$

The polynomial form:-$y_n = \sum_{i=0}^{\infty} r_{n,i}\,(x+\sigma_0)^i$

The Hypergeometrical form :- $y_n = \frac{(1+\alpha)}{\tau_1}\,{}_1F_1\big(-n; \alpha+1; \tau_1(\sigma_0+x)\big)$ and

The orthogonality integral :- $\int_{-\sigma_0}^{\infty} w\, y_n y_m dx = \frac{\Gamma(n+\alpha+1)n!exp(l\tau_1)}{(-\tau_1)^{2n+\alpha+1}}\delta_{n,n}$

This corresponds to Laguerre polynomials

Case 3

Since σ is a second degree polynomial, its two roots can be real (which may be equal or unequal) or complex conjugates. This give raise to three sub classes with the unequal roots case further divided into two subclass. Here only one of the two subclasses corresponding to the two unequal roots case is listed. This class has the attractive feature that the x range can easily be changed to [-1,1] and is widely used in numerical computations.

With l and u defined as σ=(x-l)(x-u), and two parameters α and β given by τ_1= α + β - 2 and τ_0 =-[lβ+uα+l+u], a positive weight can be obtained as:-

w=$(x-l)^\alpha(u-x)^\beta$ (here l<x<u and (α+1)(β+1)<0 ensures positive weight).

The corresponding differential equation is:-

$$(x-l)(u-x)y_n^{!!} + [(\alpha+\beta+2)x - (\alpha+1)u + (\beta+1)l]y_n^{!} = n(n+\alpha+\beta+1)y_n$$

The recursion coefficients are

$$b_{n+1} = \frac{2n(n+\alpha+n+1)(\alpha+\beta) + [l(\beta+1) + u(\alpha+1)](\alpha+\beta)}{(2n+\alpha+\beta)(2n+\alpha+\beta+2)}; c_{n-1} = \frac{n(n+\alpha)(n+\beta)(n+\alpha+\beta)(u-l)}{(2n+\alpha+\beta-1)(2n+\alpha+\beta)^2(2n+\alpha+\beta+1)}$$

When expanded around the singularity at l and u, y_n respectively have the form:-

$y_n = \frac{(l-u)^n(\alpha+\beta)_n}{(n+\alpha+\beta+1)_n}\,{}_2F_1\left(-n, n+\alpha+\beta+1; \alpha+1; \frac{(x-l)}{(u-l)}\right)$

$y_n = \frac{(u-l)^n(\beta+1)_n}{(n+\alpha+\beta+1)_n}\,{}_2F_1\left(-n, n+\alpha+\beta+1; \beta+1; \frac{(x-u)}{(l-u)}\right)$

The orthogonality integral has the form

$$\int_l^u w\, y_n y_m dx = \frac{(n!)\Gamma(n+\alpha+\beta+1)\Gamma(n+\alpha+1)\Gamma(n+\beta+1)}{\Gamma(2n+\alpha+\beta+1)\Gamma(2n+\alpha+\beta+2)}(u-l)^{2n+\alpha+\beta+1}\delta_{n,n}$$

This defines the Jacobi polynomials (after the range of x is changed to [-1,1]) and several others for special values of α & β; Chebyshev polynomial (α=β=−1/2) for example.

Another distinct roots case with range [u,∞] also exists[11a].

The two equal and complex conjugates roots cases leads respectively to Bessel and pseudo Jacobi polynomials [11a].

These are the list of classical continuous orthogonal polynomials

IV. Discrete Orthogonal Polynomials On An Equal Spaced Grid.

When the grid is equally spaced, it can be readily checked that an orthogonal set with unit weight satisfying a recursion relation can be built as follows:-

$p_0 = 1; p_1 = x_i - b_1$

$$P_{n+1} = (x_i - b_n)P_n - c_n P_{n-1}$$

with $b_{n+1} = [\sum_{i=0}^{N-1} x_i\, p_n^2] \div [\sum_{i=0}^{N-1} p_n^2]$ and $c_{n+1} = [\sum_{i=0}^{N-1} p_n^2] \div [\sum_{i=0}^{N-1} p_{n-1}^2]$ (for each x_i, x_i= x_0+hi, h, the step seize; h=(x_{i+1}-x_i); 0 ≤ i ≥ N-1). Also reduction of {9} to the form of {8} with an accuracy up to second order in h (x_{i+1}-x_i=h (i=0,1----N-1.) is possible [15] when N is sufficiently large because:-

$\frac{\delta y}{\delta x} = \left[\frac{y(x+h)-y(x)}{2h} - \frac{y(x)-y(x-h)}{2h}\right]$ and $\frac{\delta^2 y}{\delta x^2} = \frac{1}{h}\left[\frac{y(x+h)-y(x)}{h} - \frac{y(x)-y(x-h)}{h}\right]$

Thus for the difference equation {9}, parallel relations can be obtained [1,11a,15], as for {8}, for the orthogonal polynomial solution, recursion, hypergeometric representation, weighted orthogonality summation etc. Considering the conditions on the roots of σ and τ for positive weights results in 12 types (Hahn, Meixner, Kravchuk or Charlier) of possible polynomials [1,11a,15]. Among these, details of only one each of the Hahn, Meixner, Kravchuk and Charlier polynomial sets that have asymptotic solutions matching the continuous case, are listed here.

1) σ of degree 0; σ_2= σ_1=0 σ_0=1 τ_1<0

$w = \frac{1}{(-\tau_1-1)^x(x+1)!}$; x=0,1,2 ……..

Recursion :- $y_{n+1} = \left(x - n + \frac{1}{\tau_1}\right) y_n - \frac{n}{\tau_1} y_{n-1}$ n=1,23 …...

The hypergeometric form is given as $y_n = \frac{1}{(\tau_1)^n}\, {}_2F_0\,(-n, -x; -; \tau_1)$

The orthogonality sum is $\sum_{x=0}^{\infty} w\, y_m y_n = n! \frac{exp\left(\frac{-1}{\tau_1}\right)}{(-\tau)^n} \delta_{nm}$

This is known as Charlie polynomials

2a) If σ is of degree one with τ_1>1 and τ_0-τ_1 τ_0=-N w is positive ($w = \frac{1}{(\tau_1-1)^x x!(N-x)!}$).

The recursion is $y_{n+1} = \left(x - \frac{n\tau_1 - n + N}{\tau_1}\right) y_n - \frac{n(1-\tau_1)(n-1-N)}{\tau_1{}^2} y_{n-1}$

The hypergeometric form is given as $y_n = \frac{(-N)_n}{(\tau_1)^n}\, {}_2F_1\,(-n, -x; -N; \tau_1) : n = 0,1, \ldots\ldots, N$

The orthogonality sum is $\sum_{x=0}^{N} w\, y_m y_n = \frac{(\tau_1-1)^{N-2n} n!}{(\tau_1-1)^{N-n}(N-n)!} \delta_{nm}$

This is known as Krawtchouk polynomials

2b) σ is of degree one with $\sigma + \tau$=μx ; 0<μ<1; x range 0 to ∞ τ_1<0; $\tau_0 > \tau_1$

The positive weight is $w = \frac{\Gamma(x+\tau_0-\tau_1)}{(1-\tau_1)^x \Gamma(x+1)}$; x=0,1,2 ……..

The recursion is $y_{n+1} = \left(x - \frac{n(\tau_1-2)+\tau_1-\tau_0}{\tau_1}\right) y_n - \left(\frac{n(1-\tau_1)(n-1+\tau_0-\tau_1)}{\tau_1^2}\right) y_{n-1}$

The hypergeometric form is given as $y_n = \frac{(\tau_0-\tau_1)}{(\tau_1)_n}\, {}_2F_1\,(-n, -x; \tau_0 - \tau_1; \tau_1)$

The orthogonality sum is $\sum_{x=0}^{\infty} w\, y_m y_n = \frac{n!(1-\tau_1)^{n+\tau_0-\tau_1}\Gamma(n+\tau_0-\tau_1)}{(-\tau_1)^{2n+\tau_0\tau_1}} \delta_{nm}$

This is known as Meixner polynomials.

3) σ is a second degree polynomial

The requirement for the existence of positive w and finite polynomial set, can be achieved by introducing two parameters α and β defined by the following relations :-

$\tau_1 = \alpha + \beta + 2; 2\delta = \alpha + N + 1; 2\eta = \beta + N + 1$ with $\delta^2 = \left(\frac{\sigma_1-\tau_1}{2}\right)^2 - \sigma_0 + \tau_0,\ \eta^2 = \sigma_1/2^2 - \sigma_0$, (α+1)>0 and (β+1)>0.

The weight function is $w = \frac{\Gamma(x+\beta+1)\Gamma(\alpha+N+1-x)}{\Gamma(x+1)n\Gamma(N+1-x)}; x = 0,1 \ldots\ldots, N$

The recursion is :- $P_{n+1} = (x_i - b_n)P_n - c_n P_{n-1}$ with

$b_n = 2n(n-1+\tau_1)\left(2 - \sigma_1 + \frac{\tau_1}{2}\right) - \left(1 - \frac{\tau_1}{2}\right)(\tau_0 - \tau_1) / \left[(2n-2+\tau_1)\left(n + \frac{\tau_1}{2}\right)\right]$ and

$c_n = \frac{4n(n-\tau_1)D_n}{4(2n-3+\tau_1)(2n-2+\tau_1)^2(2n-1+\tau_1)}$ where $D_n = [(n-1+\tau_1/2)^2 - \delta^2 - \eta^2]^2 - 4\delta^2\eta^2$

The hypergeometric form is $y_n = \frac{(\beta+1)_n(-N)_n}{(n+\alpha+\beta+1)_n}\, {}_3F_2\,(-n, n+\alpha+\beta, -x; \beta+1, -N; 1)$

The orthogonality summation is $\sum_{x=0}^{N} w\, y_n y_m = \frac{\Gamma(n+\alpha+1)\Gamma(n+\beta+1)\Gamma(n+\alpha+\beta+N+2)n!}{\Gamma(2n+\alpha+\beta+1)\Gamma(2n+\alpha+\beta+2)(N-n)!} \delta_{nm}$

These define the Hahn polynomials.

It may be noted that for each member of the polynomial set an additional expression for the orthogonality summation exists as mentioned earlier.

At very large values of N (asymptotic form) these polynomials corresponds to the continuous version as:- Charlier & Kravchuk → Hermite, Meixner → Laguerre and Hahn → jacobi. Like Chebyshev polynomials, Hahn polynomials for the special case of $\alpha = \beta = 1/2$, referred to as Gram polynomials [25a-b]. Gram polynomials offers highest computational advantage because b_n is zero, w is unity and its zeros/roots are known.

The orthogonal polynomial parameters of all polynomials are available[15, table 2.1, 2.2 & 2.3, 23] . Also slandered expressions in terms of hyper geometrical series are available for the entire class [26].

V. Further Extensions Of Orthogonality And Orthogonal Polynomials.

Simultaneous use of P_n and its derivatives in the orthogonality integral, for better constraints, accuracy and faster convergence is routine [27a-c]. The q-analogue [11a-c] case extends the analysis to more general form of orthogonal polynomials and has advantages for the analytical approaches to solve a variety of problems. Tinkering the weight functions and/or using polynomial sets based on eigen functions of higher (even) power differential equations leads to several non-classical orthogonal functions [22]. The analysis can also be extended to multi variables and generalized cases [28,29,30,31] and find wide applications. There are several ways of generating multi variable orthogonal polynomial sets of various structures and computational complexity. Here only a set that is linked with two variable polynomial least square fit is discussed because of its use in image analysis [32].

Two dimensional orthogonal polynomials.
A N and M grid along the x and z directions respectively and a value (y_{ij})at each grid point define an image / surface and a two variable polynomial [$O_n(x,z)$] of order n over a rectangular grid of seize NxM can be used to represent it in the least square sense. For such a two variable case, the set {$x^i z^{j-i}$} ([i,j]=[n,0],[n-1,1]…..[0,n] for any n) are linearly independent for each power n. Thus a linear combination of these can replace each power of x in $O_i = \sum_{j=0}^{i} k_{i,j}\, x^j$ (see section 1.4) and via column operations, each element can be converted to a polynomial in x & z. Thus a polynomials of various degree n, $y_n(x,z)$ ($y_n(x_k, z_l) = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} r_{nij}\, O_i(x_k z_l)$), can be least square fitted to a grid with N values along x and M along z analogous to {7} as:-

$$\begin{vmatrix} \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} O_0\,(x_i z_j) y_{ij} \\ \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} O_1\,(x_i z_j) y_{ij} \\ . \\ \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} O_n\,(x_i z_j) y_{ij} \end{vmatrix} =$$

$$\begin{vmatrix} \sum_{i=0}^{N-1}\sum_{j=0}^{M-1}[O_0(x_i z_j)]^2 & \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} O_0\,(x_i z_j) O_1(x_i z_j) & . & \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} O_0\,(x_i z_{M-1}) O_{n-1}(x_i z_j) \\ \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} O_0\,(x_i z_j) O_1(x_i z_j) & \sum_{i=0}^{N-1}\sum_{j=0}^{M-1}[O_1(x_i z_j)]^2 & . & \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} O_1\,(x_i z_j) O_{n-1}(x_i z_j) \\ . & . & . & . \\ \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} O_0\,(x_i z_j) O_n(x_i z_j) & \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} O_1\,(x_i z_j) O_n(x_i z_j) & . & \sum_{i=0}^{N-1}\sum_{j=0}^{M-1}[O_n(x_i z_j)]^2 \end{vmatrix} \begin{vmatrix} r_{n,0} \\ r_{n,1} \\ . \\ r_{n,n} \end{vmatrix}$$

----------------{28}

If $O_n(xz) = P_n(x) P_n(z)$ (with $P_n(x)$ and $P_n(z)$ belonging to orthonormal polynomial sets {p_N} and {p_M} (i.e. $\sum_{i=0}^{N} w(x)\, P_k^i P_l^i = h_k \delta_{kl}$; $\sum_{k=0}^{N} w(x) \frac{P_k^i P_k^j}{h_k} = h_i \delta_{ij}$;$\sum_{i=0}^{M} w(z)\, P_k^i P_l^i = h_k \delta_{kl}$ and$\sum_{k=0}^{M} w(z) \frac{P_k^i P_k^j}{h_k} = h_i \delta_{ij}$) then, the discrete orthogonality approximated intensity represented by an n^{th} degree $O_n(xz)$ at each grid point (k,l) can be expanded over the roots of {p_N} and {p_M} as:-

$y(x_k, z_l) = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} r_{nij}\, P_i(x_k) P_j(z_l)$ ------------------------{29}

[with $r_{nij} = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} y_{ij}\, w(x) w(z) P_n(x_i) P_n(z_j)$ (for each trial n)]

This is most popular 2d construction of orthogonal polynomial with application to image analysis.[32] There are alternate approaches [33] also.

When x_1 and x_2 are two orthogonal variables with the same range, the polynomial expression, $K_n(x_1, x_2) = \sum_{k=0}^{n-1} O_i\,(x_1) O_i(x_2); (n < N+1)$ and its inverse (with $x=x_1=x_2$) are respectively known as Christoffel–Darboux (n,n) bi-degree kernel and Christoffel function. These in conjunction with polynomials built with ordered powers of two variables as basis, with some empirical modifications, have wide applications in data fit and classification of 2d data.[34,35]

VI. Orthogonal Polynomials And Quadrature

The Riemann definition of the definite integral of a function (f) yields it as the sum of the (N+1) rectangular areas $\sum_{i=1}^{N+1}(x_{i+1} - x_i)\,(y_{i+1} + y_i)/2$ (with x_i, sufficiently close (N+1) grid points at which f is y_i). Thus the form $\sum_{i=1}^{N+1} g_i\, y_i + E$ should be sought for$\int_l^u f\, dx$. Here g_i is referred to as weight, E is an error term that depends on N+1 and the functional form of f between the grid points. If indeed the integral has the form $\sum_{i=1}^{N+1} g_i\, y_i + E$ one should aim to evaluate the g_i that minimize E with a feasible, large N. An important advance is the replacement of f by its polynomial approximation of degree n. Then a direct integration of $\int_l^u f\, dx$ is possible and results in a set of n equations, $\sum_{j=0}^{n} k_{n,j} \int_l^u x^j\, dx$=$\sum_{i=1}^{N+1} g_i\, y_i$. Thus a set of N+1 g_i that is compatible with the n and $k_{n,j}$ can be obtained. Since in the above relation RHS can at most have, 2N+1 parameters, this approach is viable for solving for the weight g_i while f has a degree n ≤ (2N+1)]. In the case of equal spaced grids, though the number of parameters in RHS reduces to N+2 such an integral can be still be evaluated in this form. If the polynomial used belongs to an orthogonal set spanned on N+1 grid points, then through a series of developments initiated by Gauss, an efficient and accurate algorithm for the evaluation of g_i is available [17,22,36,37,38]. This requires that the orthogonal polynomial p_N with its roots at the grid, belongs to a finite set

$\{p_N\}$. The limits of integration need to match with the domain of the orthogonal polynomial set $\{p_N\}$ chosen (achievable via a linear transformation). From the procedure used to construct Lagrange interpolating polynomial (section 1) it is obvious that any (orthogonal) polynomial, p_N, that approximate f (with $y_i = f(z_j)$) also can be constructed from the its roots, (z_i) as:-

$$p_N = \sum_{j=1}^{N+1} \frac{L_{N+1}}{l^j(x - x_j)} y_j$$

Thus, $\int_l^u f\, dz = \int_l^u p_N\, w dz = \sum_{i=1}^{N+1} y_i \int_l^u \frac{L_{N+!} w dz}{l^j(z-z_j)} \Rightarrow g_i = \int_l^u \frac{L_{N+1} w dz}{l^j(z-z_i)}$ ---------------------{30}

This expression is valid for any orthogonal polynomial set. By noting that a suitable non unity values of the recursion coefficient, a_n, will convert this orthogonal (p_N) to an orthonormal (O_N) polynomial set and using an integrated Christoffel-Darboux identity, the above expression for g_i takes the form:-

$$g_i = \left[\frac{1}{a_n}\left(O_{N-1}(z_i) O_N^l(z_i)\right)\right]^{-1}$$

or in the more easily compute-able from:-

$\frac{1}{g_i} = \sum_{j=0}^{N} \left[O_j(z_i)\right]^2$ --------{31}

Grid points other than zeros of p_N / O_N (both have the same set of roots) can be used, as in the case of Chebyshev polynomial, where the points at which the extremes in M_N occur is used. Such cases are suitable where some boundary conditions need to be applied at the endpoints. In general, the form of the quadrature formula depends on the particular orthogonal polynomial and grids employed (needed to be found for unequal spaced grid). The Golub and Welsch [22, 36, 37] algorithm yields the grid points (as the eigen values of Jacobian, J [see equation {24}]), and also the eigen vectors. Faster algorithms are now available for this [38, 39]. Normally equal spaced grid is a voided because of the error introduced due to Runge phenomenon.
There are several means to improve accuracy and convergence [40a-c].

VII. Orthogonal Polynomials And Data Fit

As is clear from equations {3}, {5} & {13}, N+1 data points generated from a complex function or measurements, can be fitted in the least square sense to any n^{th} degree [$N \leq n \geq 0$] member of the orthogonal polynomial set $\{p_N\}$ in a non-iterative and a trial polynomial free-way provided the N data points are the roots of p_N. A linear transformation that set b_n to zero boosts the computational efficiency (because of the symmetry relations for $p_j(x)$ and$p_j(-x)$).

While approximating, higher n do not mean higher precision due to overflow / underflow (in parameters in the recursion relation and in higher powers of x_i), the lack of uniform convergence for P_n etc (for example for Jacobi with α=β=0,when N<2n(n+1, there is no uniform convergence [41]). This, in the discrete case, is assumed to contribute towards the wide swings in the computed values close to the boundary. However no such swings are prominent in the continuous case and it is conjectured that the densely spaced roots near the boundary, constrains the computed function much better. This has led to the use of additional / fake nods near the boundaries with y values matching the gradient of the data, in the discrete case. However to meet the boundary condition, fake nodes with y=0 is a better option and was investigated here.

Least square fit; computational aspects.

The data fit starts with choosing an appropriate orthogonal polynomial set. The requirement that x_i corresponds to the roots of a (N) degree orthogonal polynomial, P_N needs to be ensured during this choice. In the discrete case, the equal spaced x_i needs to be matched with the roots of P_N (z with range [-1,1] only considered here and hence a linear transformation $z_i = \epsilon_1 x_i + \epsilon_0$; $\epsilon_1 = 2/[x_N - x_1]$; $\epsilon_0 = -[x_N + x_1]/[x_n - x_1]$ is required which may be inbuilt in the recursion). In the continuous case, the y_i, that corresponds to z_{ri} is to be generated (using the function if it is being approximated or by extrapolation of data (used here) that is normally available at equal spaced grid). In both cases, closed form expressions are available for a_n, b_n and c_n for classical orthogonal polynomials. Thus for a j-degree polynomial approximation, a_j, b_j and c_j can be precomputed while computing p_{j-1} in a loop starting from 0 to j. In all such loops, orthogonal polynomial of degree j ($j \leq N-1$) are obtained (for each z_i) by employing recursions relations, initiated by appropriate value of p_0 (an inner loop over 0 to N-1). Also in same inner loop, $\left[\sum_{i=0}^{N-1} y_i\, p_j^i\right]$and $\left[\sum_{i=0}^{N-1} p_j^i\, p_j^i\right]$ ($p^i$is the value of p_j at the root z_i, of p_N) and hence $r_{j,i}$ ($r_{j,i} = \left[\sum_{i=0}^{N-1} y_i\, p_j^i\right] / \left[\sum_{i=0}^{N-1} p_j^i\, p_j^i\right]$) and the approximate y_i, $y_{j,i}$, ($y_{j,i} = \sum_{i=0}^{n} r_{j,i}\, p_j$) and root square deviation $R_j = \sum_{i=0}^{N-1}(y_i - y_{ji})\,(y_i - Y_{ji})$can be calculated.

Data fit with constant weight orthogonal polynomials

Hahn [42] and Chebyshev polynomials(T) [43] are widely used for data fit and image analysis. Two forms of Hahn polynomials with unit weight, (the Gram polynomial (G) with in the interval [-1,1] [25a-b] and a more general form (H) in the interval [x_1,x_N], [44]) and Chebyshev polynomial (T) in the interval [-1,1], are investigated. For Gram polynomial, in the standard form, the grid ($z_i = -1 + (2i-1)/N\ N \le i \ge 1$) is an equal spaced one with in the interval [-1,1]. However in this form, the grid does not include -1 & 1. A linear transformation x= (zN)/(N-1) will yield a grid ($x_{i+1} = -1 + \frac{2i}{(N-1)};\ (N-1) \le i \ge 1$) that include -1 &1 (same as in T) and is used here. For the Chebyshev polynomial, T_N, (Jacobi with $\alpha = \beta = -1/2$, b_n=0, $w = (1-z^2)^{-1/2}$) the roots (z_{ir}) are known apriori ($z_{ir} = -cos\left[(i-1/2)\frac{\pi}{N}\right]\ N \le i \ge 1$) and w is unity [43]. The grid and recursion used for each polynomial (p) is listed in table I.

As a general case, nearly equal spaced data is used and it is converted to the required (equal or at the roots of T) spaced one via interpolation. For G, the generated equal spaced data (y_i) in the x grid corresponds to the z grid (via a linear transformation $z_i = \epsilon_1 x_i + \epsilon_0\ \epsilon_1 = 2/[x_N - x_1]\ \epsilon_0 = [x_N - x_1]/[x_n - x_1]\ N \le i \ge 1$) and can be directly used. For H no such transformation is needed. For Chebyshev polynomial, the data at the Chebyshev grid ($z_{ir} = -cos\left[(i-1/2)\frac{\pi}{N}\right];\ N \le i \ge 1$) needed to be generated. For this, first a transformation of the data coordinate x_i, to z_i is carried out. Then, for getting the y_j that corresponds to each z_{ir}, initially the set of three z_j that are nearest to z_{ir} is located. Then the y_i that corresponds to z_{ir} is obtained from a parabolic interpolation that employs this set of three z_j & y_j. Thus as a general case, almost equal spaced data, can be

Table I. The grid and recursion relations used to compute the orthogonal polynomials. The subscripts o and m respectively imply orthonormal and monic forms. However, Gm is monic only in the standard grid.

P	Grid	Recursion
Go	$x_{i+1} = -1 + \frac{2i}{(N-1)};\ (N-1) \le i \ge 0$	$g_n = a_{n-1}\frac{N-1}{N} z g_{n-1} - \frac{a_{n-1}}{a_{n-2}} g_{n-2}$ $g_0 = 1$ and $a_{n-1} = \sqrt{\left(\frac{4-1/n^2}{1-(n/N)^2}\right)}$
Gm	$x_{i+1} = -1 + \frac{2i}{(N-1)};\ (N-1) \le i \ge 0$	$g_n = \frac{N-1}{N} z g_{n-1} - c_{n-1} g_{n-2}$ $g_0 = 1$ and $g_1 = z$ $c_n = \left(\frac{1-(n/N)^2}{4-1/n^2}\right)$
Ho	$x_i = s(i-1) - (x_n - x_1)/2\ (N-1) \le i \ge 1$ $s = x_{i+1} - x_i; (N-1) \le i \ge 1$	$a_{n+1} = \frac{2}{hn}\sqrt{\frac{(2n+1)(2n-1)}{(N^2-n^2)}}$ $b = (x_N + x_1)/2\ c_n = \frac{a_n}{a_{n-1}}$ $h_n = a_n[x-b]h_{n-1} - c_n h_{n-2}$ n=3,4,----N $h_1 = \frac{1}{\sqrt{N}};\ h_2 = a_2(x-b)h_1$
Hm	$x_i = s(i-1) - (x_n - x_1)/2\ (N-1) \le i \ge 1$ $h = x_{i+1} - x_i; (N-1) \le i \ge 1$	$b = (x_n + x_1)/2\ \ c_{n+1} = \left(\frac{s}{2}\right)^2 \frac{(N^2-n^2)}{(4-1/n^2)}$ $h_n = [x-b]h_{n-1} - c_n h_{n-2}$ n=3,4,----N $h_1 = 1\ \ h_2 = x - b$
To	$-cos\left[(i-1/2)\frac{\pi}{N}\right];\ N \le i \ge 1$	T_0=1 & T_1=z T_{n+1}=2zT_{n-1} - T_n n ≥ **1**
Tm	$-cos\left[(i-1/2)\frac{\pi}{N}\right] N \le i \ge 1$	T_0=1 , T_1=z and T_2=zT_1-T_0/2 T_{n+2}=zT_{n+1}-T_n/4 ; n ≥1

approximated to polynomials of various degrees as $\left(f_j = \sum_{i=0}^{n} r_{j,i}\, p_i\right)$ by evaluating $r_{j,i}$ as:-$r_{j,i} = \left(\sum_{i=0}^{N-1} y_i\, {p^i}_j\right) \div \left(\sum_{j=0}^{N-1} {P^i}_j\, {P^i}_j\right)$. The polynomial form of f_n is can be computed ($f_j = \sum_{i=0}^{j} r_{j,i}\, p_i$=$\sum_{i=0}^{j} r_{j,i} \sum_{j=0}^{i} k_{i,j}\, z^j$=$\sum_{i=0}^{j}\left(r_{j,i}\sum_{j=0}^{i} k_{i,j}\right) z^i$) using the $r_{j,i}$ and k_{ii} obtained from the recursion relation {17b}**.** However this will be for z which has the range [-1,1] (for G & T). To get the polynomial form in the original range [x_1,x_N], the recursion for k_{iji} corresponding to the modified recursion (equation{19}) for the orthogonal polynomial as per the liner transformation ($x_i = \eta_1 z_i + \eta_0;\ \eta_1 = \frac{2.}{(x_N - x_1)};\ \eta_0 = \frac{-(x_N + x_1)}{(x_N + x_1)}$) needs to be used. The, modified recursion of the orthogonal polynomial and the corresponding recursion for $k_{i,j}$ are listed in table II. Using the recursion to generate the polynomial coefficients is less cumbersome as compared to the alternate method of using binomial coefficients$\binom{n}{k}$.

VIII. Data Fit And Results

Traditionally, for stable computation, lot of emphasis is placed on a suitable starting (scaled) value of the orthogonal polynomial and the computation of recursion coefficients, to avoid over/under flow for large values n<N-1 [42]. In the classical orthogonal polynomial, constant-weight, cases used here, the coefficients of the recursion relation do not diverge and hence any computational error should arise from that of higher powers of x or other sources. Thus with the standard starting values given above, the orthogonal polynomials of various n values (0 >n< N) were computed (employing GFORTRAN compiler with quadrupole precision). Two Planck profiles corresponding to nearly equal spaced (0.5nm), 10000 discrete x values, in the range [350,5350], one with added random counts to mimic noise and another without any noise, were generated and fitted to orthogonal polynomial after converting the data to required steps (to equal steps or at the roots of T_N). To evaluate the quality of data fit and noise rejection, the of sum of square of deviation (R) for each degree (n) up to 9999 was computed (unless limited by overflow).

Table II Recursion for the polynomial coefficients $k_{j,i}$ for each n.

p	Transformation	Recursion
Go	$x_i = \eta_1 z_i + \eta_0$; $\eta_1 = \frac{2.}{(x_N - x_1)}$; $\eta_0 = \frac{-(x_N + x_1)}{(x_N + x_1)}$	$g_n = (\eta_1 x + \eta_0) g_{n-1} - c_{n-1} g_{n-2}$
		$k_{0,0} = 1; k_{(1,0)} = \eta_0; k_{1,1} = \eta_1$ $k_{(i,j)} = \eta_1 k_{i-1,j-1} - \eta_0 k_{i-1,j} - r_{n,i-2} k_{i-2,j}; 2 \le i \ge n; j \le i$
Gm	$x_i = \eta_1 z_i + \eta_0$ $\eta_1 = \frac{2.}{(x_N - x_1)}$; $\eta_0 = \frac{-(x_N + x_1)}{(x_N + x_1)}$)	$g_n = a_{n-1}(\eta_1 x + \eta_0) g_{n-1} - \frac{a_{n-1}}{a_{n-2}} g_{n-2}$
		$k_{0,0} = 1; k_{(1,0)} = \eta_1 r_0; k_{1,1} = \eta_1 r_{n,0}$ $k_{(i,j)} = \eta_1 r_{n,(i-1)} k_{i-1,j-1} - r_{n,(i-1)} \eta_0 k_{i-1,j} - r_{n,(i-1)} k_{i-2,j} / r_{n,(i-2)}; 2 \le i \ge n; j \le i$
Ho		$h_n = a_n [x - b] h_{n-1} - c_n h_{n-2}; \quad b = (x_n + x_1)/2$
		$k_{0,0} = 1; k_{(1,0)} = b k_{0,0}; k_{1,1} = 1$ $k_{(i,j)} = [k_{i-1,j-1} + b k_{(i-1,j)} - k_{i-2,j}] r_{n,i}; 2 \le i \ge n; j \le i$
Hm		$h_n = [x - b] h_{n-1} - c_n h_{n-2}$; $b = (x_n + x_1)/2$
		$k_{0,0} = \sqrt{1/N}; k_{(1,0)} = r_2 b k_{0,0}; k_{1,1} = r_{n,2}$; $k_{(i,j)} = r_{n,(i+1)} \left[k_{i-1,j-1} + b k_{(i-1,j)} - \frac{k_{i-2,j}}{r_{n,i}} \right]; 2 \le i \ge n; j \le i$
To	$x_i = \eta_1 z_i + \eta_0$; $\eta_1 = \frac{2.}{(x_N - x_1)}$; $\eta_0 = \frac{-(x_N + x_1)}{(x_N + x_1)}$	$T_{n+1} = 2(\eta_1 x + \eta_0) T_{n-1} - T_n$
		$k_{0,0} = 1; k_{(1,0)} = \eta_0; k_{1,1} = \eta_1$; $k_{(i,j)} = 2\eta_1 k_{i-1,j-1} + 2\eta_0 k_{(i-1,j)} - k_{i-2,j}; 2 \le i \ge n; j \le i$

Polynomials $f_n = \sum_{i=0}^{n} k_{n,i} x^i$ in the range [x_1,x_N] (=[350,5350])were computed using Horner recursive algorithm [21] (referred to as converted polynomial, subsequently) after obtaining $r_{n,i}$ & k_{ni}. In all cases (with and without noise for all orthogonal polynomials and converted polynomials), the root mean square deviation (R) was computed to identify convergence and sign of overflow/underflow. The results are summarized for data with and without added noise (RMS value of noise added is 5907) in Table T00. The R&n value at the highest computable n for orthogonal polynomial (R_s^o) , the R minimum and the corresponding n for orthogonal and converted polynomials respectively ($R_{sm}^o$and R_{sm}^p) and the R&n value of orthogonal polynomial at the n value at which the converted polynomial has the minimum (R_s^{op}) are listed. The subscript s is replaced by n for data to which noise is added. The executable versions (ubuntu25.4) of data fitting codes and data generation code are available in the link:- https://drive.google.com/drive/folders/10P0Bf4remlCH0wZsiWPtAv18vfAWKy9A?usp=sharing

Table T00. Listed are the degree of the polynomial along with the root mean square deviation (with in brackets) for fits with Chebyshev (T), monic Gram (Gm), orthogonal gram(Go), monic (Hm) and orthogonal (Ho) polynomials (o). Also listed are the R values of the polynomial (p) in the original x values obtained by the appropriate recursion relation (only the degree at which the minimum in R occur is listed). R have respective subscript s or n (Rs &Rn) without and with noise (the RMS value of noise added to the signal is 5907). An additional subscript m ($R_{sm}^o R_{nm}^o R_{sm}^p R_{nm}^p$) implies that R has the lowest value for the degree given.

R	Polynomial				
	T	Gm	Go	Hm	Ho
R_s^o	**9999(0)**	9999 (3.48 x10^7)	9999(3.29x10^7)	1585 (6.55)	9999(3.26x10^7)
R_n^o	**9999(0)**	9999(3.51x10^7)	9999(3.31x10^7)	1585(5277)	9999(3.28x10^7)
R_{sm}^o	**9999(0)**	1117(5.51)	1117(5.51)	792(6.56)	1114(5.51)
R_{nm}^o	**9999(0)**	1188(5163)	1196(5160)	792(5280)	1158(5173)
R_{sm}^p	**43(16.2)**	**43(16.4)**	**44(16.4)**	**44(16.4)**	**45(16.4)**
R_{nm}^p	**43(5210)**	**42(5563)**	**46(6956)**	**43(5563)**	**42(5563)**
R_s^{op}	**43(16.2)**	**44(16.4)**	**44(16.4)**	**45(16.40)**	**45(16.4)**
R_n^{op}	**43 (5210.8)**	**43(5563)**	**46(5561)**	**44(5563)**	**43(5563)**

For the Chebyshev (T) polynomial R decreases continuously from the n value of zero to N-1 without any sign of over/under flow. However, the converted polynomial coefficients of higher power of x could be computed only to around a degree of forty three due to underflow of the polynomial expansion coefficient k_{nj}. This n at which R have the minimum (listed in R_{sm}^{p} and R_{nm}^{p}) can be identified as the best orthogonal polynomial degree to be used for data fit with optimum noise rejection. This is obvious in the data presented in the table T00 where it can be observed that at n for which R has the lowest value, it is closest to the RMS noise added (listed in R_{sm}^{op} and R_{nm}^{op} for data with and without noise). In computations with data to which noise is added, a match between the simulated and computed data is obtained with orthogonal polynomials (as expected) for sufficiently large n. This imply that T have lower noise discrimination at higher degree and can even fit a noisy data as by design they should pass through all points.

In the discrete cases, as seen in table T00, the general trend in the case of orthogonal polynomial is for R to fall and then start increasing resulting in a minimum (R_{sm}^{o} R_{nm}^{o}). However, except for Hm (where sudden underflow occurred at low n), it was possible to compute up to n close to N-1(though R is large). The R value for the converted polynomial (R_{sm}^{p} & R_{nm}^{p}) also had a minimum due to underflow of the polynomial expansion coefficient k_{nj}. The R value at which its minimum occurs is larger for fit to orthogonal polynomial as compared to the converted polynomial (R_{sm}^{o}, R_{sm}^{p}, R_{nm}^{o} and R_{nm}^{p}).

Monitoring the computed orthogonal polynomial for various grid points showed that the R_{n}^{o} & R_{s}^{o}start increasing from its minimum due to spike in computed values close to the boundary. When n approaches N-1, these spikes in the few computed values close to the x limits are large resulting in huge values of R_{s}^{o} & R_{n}^{o}. R computed avoiding five values (there is no spikes for other x values) close to the boundaries, yields a R that is only marginally larger than its minimum value, even when n approach N. This indicates that it is for x values close to the limits that the computational instability occurs and imposing boundary conditions via fake nodes [35], is the cure for it. Zero counts in three fake nodes, the boundary condition (both the polynomial and its derivative zero at the boundary) required for orthogonality of the discrete orthogonal polynomials [11a] will be satisfied. The effect of fake nodes (results are tabulated for five nodes, though in trials it is seen three is enough and above ten beneficial effect deteriorate) that will enforce the required boundary condition in the discrete case was investigated. For Gm, Go, Hm and Ho polynomials, in T50 (all 5 fake boundary values zero, thus function and its derivatives zero at the boundary) and T51(all boundary values are same as the original; thus function is constant and its derivative zero at the boundary), R values are listed.

Table T50. R values with five fake nodes of zero values. There is no divergence at higher degree orthogonal polynomials (R_{s}^{o} R_{n}^{o}) in gm & go. For hm there is not much improvements. The initial convergence is poorer for both converted and orthogonal polynomials.

R	Polynomial			
	Gm	Go	Hm	Ho
R_{s}^{o}	**9999(5.16)**	**9999(5.17)**	**1586 (6.5)**	**9999(5.22)**
R_{n}^{o}	**9999(5154)**	**9999(5152)**	**1586(5284)**	**9999(5161)**
R_{sm}^{o}	1254(5.16)	19(6721)	797(6.56)	1228(5.22)
R_{nm}^{o}	1215(5156)	19(8731)	797(5281)	1215(5159)
R_{sm}^{p}	18(6787)	19(16285)	44(12775)	44(15338)
R_{nm}^{p}	18(8770)	45(5151)	44(12352)	44(12183)
R_{s}^{op}	18(15920)	19(6721)	44(11047)	44(10838)
R_{n}^{op}	18(16828)	45(10046)	44(12306)	44(12306)

Table T51 Computed R values with five fake nodes in which boundary values are repeated. The divergence at higher degree orthogonal polynomials (R_{s}^{o}) in gm & go is due to the noncompliance of the boundary condition and in hm & ho this is further aggravated by the extended range of the variable.

R	Polynomial			
	Gm	Go	Hm	Ho
R_{s}^{o}	9999(3.43 x10^{7})	9999(10924)	1585(6.56)	9999(3.2x10^{7})
R_{n}^{o}	9999(3.46x10^{7})	9999(12200)	1585(5281)	9999(3.3x10^{7})
R_{sm}^{o}	33(21.03)	35(18.18)	797(6.56)	1105(5.51)
R_{nm}^{o}	48(5562)	32(5568)	797(5281)	1156(5176)
R_{sm}^{p}	**34(18.19)**	**44(19.89)**	**44(22.58)**	**44(20.4)**
R_{nm}^{p}	**31(5568)**	**44(5584)**	**43(5565)**	**44(5569)**

R_s^{op}	**33(20.7)**	**44(18.13)**	**44(19.8)**	**44(19.86)**
R_n^{op}	**48(5562)**	**44(5562)**	**44(5565)**	**44(5564)**

In T50 it is seen that all except Hm, are computable to the highest possible degree with R values close to its minimum (no divergence). However, it is noted that the initial convergence rate is smaller now as compared to T00 case. The convergence rate of the converted polynomial is smaller and the minimum R value at which it starts diverging is larger as compared to T00 case.

The performance of Gm & Go differ from that of T though, in all, the range of x is [-1,1]. Though, computations up to highest power of orthogonal polynomial is possible in T without any fake node, the noise is discriminated only at lower degree. In Gm and Go with fake nodes, even at the highest polynomial degree, the noise appears to be discriminated (T50). The reason for the sudden divergence of Hm is not clear.

In T51 it is seen that the convergence rate of all are almost the same and larger (minimum in R occurs at lower degree) than in the T00 case. However the orthogonal polynomial diverges at higher n, while the converted polynomial converges faster with R minimum occurring at lower n. Thus for data fit to a polynomial of lowest degree, Gm is better than other cases.

IX. Conclusion

With appropriate boundary conditions (as in T50), the divergence in the computation of orthogonal polynomial can be avoided for data on equal spaced grid. For data fit to lowest degree polynomial with maximum noise discrimination, Gm with three or more fake nodes with boundary values repeated (as in T51) is optimum.

References

[1]. Arnold F. Nikiforov And Vasilii B. Uvarov; Special Functions Of Mathematical Physics;1988; Springerbaselag.
[2]. Fernando Albiac, José L. Ansorena And Vladimir Temlyakov; Twenty-Five Years Of Greedy Bases; J. Approximation Theory 2025; 307;106141; Https://Doi.Org/10.1016/J.Jat.2024.106141)
[3]. Pham-Gia, T. And Thanh, D.N; Hypergeometric Functions: From One Scalar Variable To Several Matrix Arguments, In Statistics And Beyond; Open Journal Of Statistics,2016; 6(5), 951 ; DOI: 10.4236/Ojs.2016.65078
[4]. Https://Mathworld.Wolfram.Com/Hypergeometricfunction.Html
[5]. Koepf, W; Hypergeometric Summation, An Algorithmic Approach To Summation And Special Function Identities; 2014;Springer, London.
[6]. Kil H Kwon And Lancel Littlei; Classification_Of_Classical_Orthogonal_Polynomials; J.Korean Mathematical Society 1997; 34(4), 1. Https://Www.Researchgate.Net/Publication/238203450_
[7]. Sury.B; Weierstrass's Theorem - Leaving No 'Stone' Unturned; Resonance 2011; 16; 341 Https://Www.Ias.Ac.In/Article/Fulltext/
[8]. Https://Mathworld.Wolfram.Com/Taylorseries.Html
[9]. Https://En.Wikipedia.Org/Wiki/Trigonometric_Polynomial
[10]. K.Srinivasa Rao And Vasudevan Lakshminarayanan; Genralized Hypergeometric Functions; IOP Publishing Ltd 2018 (Chapter1) (Pdf)
[11]. Katherine N Quinn Michael C Abbott, Mark K Transtrum, Benjamin B Machta And James P Sethna; Information Geometry Of Multi Parameter Models: New Perspectives On The Origin Of Simplicity; Rep. Prog. Phys. 2023; 86, 035901 ; DOI 10.1088/1361-6633/Aca6f8 ; (Pdf)].
[12]. Cristina Cornelio, Sanjeeb Dash, Vernon Austel, Tyler R.Josephson, Joaogoncalves, Kenneth Clarkson, Nimrod Megiddo, Bachir El Khadir And Lior Horesh; AI Descartes: Combining Data And Theory For Derivable Scientific Discovery; Nat Commun 2023; 14; 1777.; Https://Doi.Org/10.1038/S41467-023-37236-Y
[13]. Mikhail Belkin; Fit Without Fear: Remarkable Mathematical Phenomena Of Deep Learning Through The Prism Of Interpolation; Acta Numerica 2021; 30, 203. ; DOI: Https://Doi.Org/10.1017/S0962492921000039 (Pdf)
[14]. Tofallis, C; Fitting An Equation To Data Impartially; Mathematics 2023; 11(18); 3957, Https://Doi.Org/10.3390/Math11183957
[15]. Https://Mathworld.Wolfram.Com/Lagrangeinterpolatingpolynomial.Html
[16]. Ahmed Arafat And Moawwad El-Mikkawy; A Fast Novel Recursive Algorithm For Computing The Inverse Of A Generalized Vandermonde; Matrix; Axioms 2023; 12; 27. ; Https://Doi.Org/10.3390/Axioms12010027)
[17]. De Marchi, S., Elefante, G. And Marchetti, F. Stable; Discontinuous Mapped Bases: The Gibbs–Runge Avoiding Stable Polynomial Approximation (GRASPA) Method; Comp. Appl. Math.2021; 40, 299 . ; Https://Www.Semanticscholar.Org/Reader /2f9a8bea242eb862ea7 10f79e780b0e8d0e63275.
[18]. Rodrigo B. Platte, Lloyd N. Trefethen And Arno B. J. Kuijlaars; Impossibility Of Fast Stable Approximation Of Analytic Functions From Equispaced Samples; SIAM Review 2011; 53; 308. DOI: 10.1137/090774707 Pdf.
[19]. R. Koekoek, P. A. Lesky, R. F. Swarttouw; Hypergeometric Orthogonal Polynomials And Their Q-Analogues; 2010; DOI 10.1007/978-3-642-05014-5; Springer Science.
[20]. Mourad E.H. Ismail, Plamen Simeonov; Q-Difference Operators For Orthogonal Polynomials; J. Computational And Applied Mathematics 2009; 233 749.; Https://Www.Sciencedirect.Com/Science/ Article/Pii/S0377042709001228
[21]. F. Marcellan And J.C. Medeem; Q−Classical Orthogonal Polynomials: A Very Classical Approach; Electronic Transactions On Numerical Analysis 1999; 9; 112, Pdf
[22]. George B. Arfken, Hans J. Weber And Frank E. Harris; Mathematical Methods For Physicists A Comprehensive Guide; Viith Ed 2013, Elsevier Inc (Chapter 8 & 12) (Pdfdrive)
[23]. Kyle R. Bryenton, Andrew R. Cameron, Keegan L. A. Kirk, Nasser Saad, Patrick Strongman And Nikita Volodin; On The Solutions Of Second-Order Differential Equations With Polynomial Coefficients: Theory, Algorithm, Application; Algorithms 2020; 13(11); 286. ; Https://Doi.Org/10.3390/A13110286
[24]. J.L. Cardoso, C. M. Fernandes, And R. Alvarez-Nodarse; Structural And Recurrence Relations For Hypergeometric-Type Functions By Nikiforov-Uvarov Method; Electronic Transactions On Numerical Analysis.2009; 35,17. Https://Www.Researchgate.Net/Publication/228573836

[25]. Macquarrie, L., Saad, N. & Islam, M.S; Asymptotic Iteration Method For Solving Hahn Difference Equations; Adv Differ Equ ;2021; 2021(1), 354 . (Pdf)

[26]. Bilender P. Allahverdiev And Hüseyin Tuna; Spectral Theory Of Singular Hahn Difference Equation Of The Sturm-Liouville Type; Communications In Mathematics; 2020; 28(1); 13; Https://Www.Researchgate. Net/Publication/340878994

[27]. A.F. Nikiforov, S.K. Suslov, And V.B. Uvarov; Classical Orthogonal Polynomials Of A Discrete Variable; Springer Series In Computational Physics, Springer-Verlag, Berlin, 1991

[28]. Claudebrezinski; A Direct Proof Of The Christoffel Identity And Its Equivalence To The Recurrence Relationship; J. Computational And Applied Mathematics 1990; 32(1-2); 17. (Pdf)

[29]. Victor Kawasaki-Borruat; Generalized Gaussian Quadrature For Integrals With Singular Weights; Thesis, (2022); Https://People.Math.Ethz.Ch/~Hiptmair/Studentprojects/Borruat.Victor/Bscthesis.Pdf

[30]. E. Godoy, A. Ronveaux, A. Zarzo, I. Area; Minimal Recurrence Relations For Connection Coefficients Between Classical Orthogonal Polynomials: Continuous Case; J Of Comput And Applied Mathematics 1997; 84(2), 257; Https://Www.Sciencedirect.Com/Science/Article/Pii/S0377042797001374?Via%3Dihub

[31]. I.Area, E.Godoy, A.Ronveaux, A.Zarzo; Minimal Recurrence Relations For Connection Coefficients Between Classical Orthogonal Polynomials: Discrete Case; J Of Comput And Appl Mathematics 1998; 89; Https://Doi.Org/10.1016/S0377-0427(98)00002-8

[32]. Wolfram Koepf, Dieter Schmersau; Representations Of Orthogonal Polynomials; J. Computational And Applied Mathematics 1998; 90; 57; Https://Doi.Org/10.1016/S0377-0427(98)00023-5

[33]. Https://Handwiki.Org/Wiki/Clenshaw_Algorithm

[34]. Roberto Barrio; Rounding Error Bounds For The Clenshaw And Forsythe Algorithms For The Evaluation Of Orthogonal Polynomial Series; J Computational And Applied Mathematics 2002; 138; 185; Https://Doi.Org/10.1016/S0377-0427(01)00382-X

[35]. Https://En.Wikipedia.Org/Wiki/Horner%27s_Method

[36]. Herbert.S. Wilf; Mathematics For The Physical Sciences; Wiley, New York, 1962. (Pdf)

[37]. Koornwinder, T.H; Orthogonal Polynomials; In: Computer Algebra In Quantum Field Theory. Texts & Monographs In Symbolic Computation; Schneider, C., Blümlein, J. (Eds); Springer, Vienna 2013; Https://Doi.Org/10.1007/978-3-7091-1616-6_6 1303.2825.Pdf (Arxiv.Org)

[38]. Https://Mathworld.Wolfram.Com/Gram-Schmidtorthonormalization.Html

[39]. R.W. Barnard; G. Dahlquist; K. Pearce; L. Reichel And K.C. Richards; Gram Polynomials And The Kummer Function J Approximation Theory 1998; 94(1) 128; Https://Doi.Org/10.1006/Jath.1998.3181

[40]. Dimitar K. Dimitrov And Lourenço L. Peixoto; An Efficient Algorithm For The Classical Least Squares Approximation; SIAM J Scientific Computing 2020; 42(5); 17; Https://Www.Researchgate.Net/Publication/334191250_

[41]. Roelof Koekoek And René F. Swarttouw: The Askey-Scheme Of Hyper Geometric Orthogonal Polynomials And Its Q-Analogue; Https://Fa.Ewi.Tudelft.Nl/~Koekoek/Askey/

[42]. Juan C. García-Ardila, And · Misael E. Marriaga ,Approximation By Polynomials In Sobolev Spaces Associated With Classical Moment Functionals; Numerical Algorithms 2024; 95; 285; DOI:10.1007/S11075-023-01572-3 (Pdf)

[43]. Francisco Marcellán And Yuan Xu; On Sobolev; Orthogonal Polynomials; Expositions; Mathematicae 2015; 33(3); 308; Https://Doi.Org/10.1016/J.Exmath.2014.10.002

[44]. Roland Ritt,Matthew Harker And Paul O'Leary; Simultaneous Approximation Of Measurement Values And Derivative Data Using Discrete Orthogonal Polynomials; IEEE International Conference On Industrial Cyber Physical Systems (ICPS), Taipei, Taiwan, 2019, Pp. 282-289, Doi: 10.1109/ICPHYS.2019.8780356. (Pdf)

[45]. A. Bultheela, A. Cuytb, W. Van Asschec, M. Van Barela And B. Verdonk; Generalizations Of Orthogonal Polynomials; J Computational And Applied Mathematics 2005; 179; 57; (Pdf)

[46]. Yuan Xu; Orthogonal Polynomials Of Several Variables; Https://Arxiv.Org/Pdf/1701.02709

[47]. Mourad E.H. Ismail And Ruiming Zhang; A Review Of Multivariate Orthogonal Polynomials; J Egyptian Mathematical Society2017; 25 (2); 91;

[48]. Plamen Iliev And Yuan Xu; Discrete Orthogonal Polynomials And Difference Equations Of Several Variables; Advances In Mathematics 2007; 212 ; 1;; Https://Arxiv.Org/Abs/Math/0508039

[49]. K.W.See, K.S.Loke, P.A.Lee And K.F. Loe; Image Reconstruction Using Various Discrete Orthogonal Polynomials In Comparison With DCT; Applied Mathematics And Computation 2007; 193; 346.

[50]. Baoxu Regularization To Orthogonal-Polynomials Fitting With Application To Magnetization Data Https://Arxiv.Org/Abs/1603.03532]

[51]. Jean Bernard Lasserre, Edouard Pauwels And Mihai Putinar; The Christoffel–Darboux Kernel For Data Analysis; Cambridge University Press 2022 Doi.Org/10.1017/9781108937078

[52]. Stefano DE MARCHI; Padua Points And Fake Nodes For Polynomial Approximation: Old,New And Open Problems; Constructive Mathematical Analysis2022; 5(1);14. Https://Www.Researchgate.Net/Publication/359127142_

[53]. Walter Gautschi; A Survey Of Gauss-Christoffel Quadrature Formulae; Https://Www.Cs.Purdue.Edu/ Homes/Wxg/Selected_Works/Section_12/074.Pdf]