*International Journal of Current Research in Multidisciplinary (IJCRM)*
*ISSN: 2456-0979*
www.ijcrm.com ‖ Volume 1 ‖ Issue 4 ‖

# Computational Aspects of q-Method

Singh Prashant[1]
[1]*(Department of Computer Science, Banaras Hindu University, India)*

***Abstract :*** *This paper contains some definitions, abbreviations, acronyms, concepts and coding of Numerical Computation involving q method. Numerical Computation plays an imperative role in solving real time and real life problems of engineering, mathematics and physics. It is an approach for solving complex mathematical problems using arithmetic operations. This approach involves formulation of mathematical models of physical situations that can be solved with mathematical operations. Applications of computer oriented numerical methods have become an integral part of life of all the modern scientists.*

***Keywords :****q method, q hyper-geometric function, basic hyper-geometric function etc.*

## I. Introduction

The field of numerical analysis predates the discovery of modern computers by many centuries. Linear interpolation was already in use for more than ten centuries. Many great mathematicians of the past were engrossed in study of numerical analysis and it is alsoapparent from the names of important algorithms like Newton's method, Lagrange interpolation polynomial, Gaussian elimination, or Euler's method.To facilitate computations by hand, large books were produced with formulas and tables of data such as interpolation points and function coefficients. Using these tables, often calculated out to sixteen decimal places or more for some functions, one could look up values to plug into the formulas given and achieve very good numerical estimates of some functions.

The mechanical calculator was also developed as anapparatus for hand computation. These calculators evolved into electronic computers in the first generation of computers, and it was then found that these computers were also useful for administrative purposes. The discovery of the computer also influenced the field of numerical analysis, since now longer and more complicated calculations could be done. Numerical computing with the help of some special functions enhanced computational techniques.

Heine's **basic hypergeometric series**, or **hypergeometric *q*-series**, are *q*-analogue generalizations of generalized hypergeometric series, and are in turn generalized by elliptic hypergeometric series. A series $x_n$ is called hypergeometric if the ratio of successive terms $x_{n+1}/x_n$ is a rational function of $n$. If the ratio of successive terms is a rational function of $q^n$, then the series is called a basic hypergeometric series. The number $q$ is called the base.The basic hypergeometric series was first considered by Eduard Heine (1846). It becomes the hypergeometric series $F(\alpha,\beta;\gamma;x)$ in the limit when the base $q$ is 1.Some of the q analogues are explained below.

### 1.1 q Analogue of Exponential Function

***q*-exponential**is a *basic* analogue [Exton(1983)] of the exponential function, namely the eigen function of a *q*-derivative. There are many *q*-derivatives, for example, the classical *q*-derivative, the Askey-Wilson [Wilson et al. (1985)] operator, etc. Therefore, unlike the classical exponentials, *q*-exponentials are not unique. Three variants of exponential functions are given below. Third one is generalized formula and we can get first and second by changing values of $\alpha$.

$$E_{q^{-1}}(x) = \sum_{r=0}^{\infty} \frac{x^r q^{r(r-1)/2}}{[r;q]!} \tag{1.1}$$

$$E_q(x) = \sum_{r=0}^{\infty} \frac{x^r}{[r;q]!} \tag{1.2}$$

$$E(q,\alpha;x) = \sum_{r=0}^{\infty} \frac{x^r q^{\frac{r\alpha(r-1)}{2}}}{[r;q]!} \tag{1.3}$$

**1.2 *q Analogue of Basic*Integration:** The inverse operation to basic differentiation has also been discussed at some length by F.H. Jackson [Jackson (1904)].This is represented by the symbol $\int_a^b \emptyset(x)\, d\,(qx)$ and is referred to as *q*-integration or basic integration. When *q* tends to unity, the basic integral reduces to the ordinary integral. The operations of basic differentiation and integration correspond exactly in every way to ordinary differentiation and integration of which these are generalizations.

$$\int_a^b f(x)d(qx) = (1-q)\{b\sum_{r=0}^{\infty} q^r f(q^r b) - a\sum_{r=0}^{\infty} q^r f(q^r a)\} \tag{1.4}$$

$$\int_0^c f(x)d(qx) = (1-q)\{c\sum_{r=0}^{\infty} q^r f(q^r c)\} \tag{1.5}$$

$$\int_{cq}^{\infty} f(x)d(qx) = (1-q)\{c\sum_{r=0}^{\infty} q^{r+1} f(q^{r+1} c)\} \tag{1.6}$$

$$\int_0^{\infty} f(x)d(q,x) = (1-q)\sum_{i=-\infty}^{\infty} q^i f(q^i) \tag{1.7}$$

**1.3 q Analogue of Trigonometric Functions :**As like exponential function, various analogues of circular functions have been introduced. Jackson gave a class of *q*-circular function[Exton (1983)], [Gasper et al. (2004)] .He also introduced a class of circular function from point of view of pseudo-periodicity.

$$sin_q(x) = x\sum_{r=0}^{\infty}\frac{(-x^2)^r}{[2r+1;q]!} = x0F1(-;\tfrac{3}{2};q^2;-\left[\tfrac{1}{2};q^2\right]^2 x^2) \tag{1.8}$$

$$cos_q(x) = \sum_{r=0}^{\infty}\frac{(-x^2)^r}{[2r;q]!} = 0F1(-;\tfrac{1}{2};q^2;-\left[\tfrac{1}{2};q^2\right]^2 x^2) \tag{1.9}$$

**1.4 Properties of Trigonometric Functions**

$$sin_q(x)sin_{1/q}(x) + cos_q(x)cos_{1/q}(x) = 1 \tag{1.10}$$

$$cos_q(x)cos_{1/q}(x) - sin_q(x)sin_{1/q}(x) = cos_q(2x) \tag{1.11}$$

**1.5 Basic Differentiation operator:** Jackson introduced the operative symbol[Exton (1983)], [Gasper et al. (2004)] for basic differentiation defined by the relation $\Delta\{\emptyset(x)\} = \{(x) - (qx)\}\, x^{-1}(1-q)^{-1}$, The operation of basic differentiation is defined by [Jackson (1904)] the relations

$$B_{q,x}\emptyset(x) = \frac{\emptyset(x)-\emptyset(qx)}{x(1-q)} = \sum_{r=0}^{\infty}\frac{(q-1)^r x^r d^{r+1}\emptyset(x)}{(r+1)!\; dx^{r+1}}, \tag{1.12}$$

, where *x* and *q* may be real or complex. This becomes the same as ordinary differentiation as the base *q* tends to unity. In order to avoid the possibility of perplexity with the ordinary difference operator, we shall write $B_{q,x}$ instead of $\Delta$. Furthermore, the subscripts *q* and *x* will be omitted provided that there is no chance of vagueness. It will be seen that the possibility now arises of the existence of certain types of difference equations based upon this operator.

$$D_{q,x}f(x)=\frac{f(qx)-f(x)}{x(q-1)} \tag{1.13}$$

**1.6 Basic analogue of Taylor's Theorem :**Jackson introduced [Exton (1983)] *q* analogue of Taylor's Theorem

$f(x) = f(a) + \frac{(x-a)^{(1)}}{[1;q]}D_q f(a) + \frac{(x-a)^{(2)}}{[2;q]!}D_q^2 f(a) + \ldots + \frac{(x-a)^{(n)}}{[n;q]!}D_q^n f(a)$, where

$R_n = \frac{(x-a)^{(n+1)}}{[n+1;q]!}D^{(n+1)}f(\xi)$, where $\xi$ lies between x and a. (1.14)

**1.7 Variants of Laplace Transform :**Hahn [Hahn (1949)] defined two analogues of Laplace Transform by the help of the integral equations

$$L_{q,s}f(x) = \frac{1}{1-q}\int_0^{\frac{1}{s}} E_q(qsx)f(x)d(q,x) \tag{1.15}$$

$$\mathcal{L}_{q,s}f(x) = \frac{1}{1-q}\int_0^\infty e_q(-sx)f(x)d(q,x) \qquad (1.16)$$

$\mathrm{Rl}(s) \geq 0$

$$L_{q,s} = \frac{1}{1-q}\int_0^{\frac{1}{s}} E_q\,(qsx)\,f(x)d(q,x) = \frac{(q,\infty)}{s}\sum_{j=0}^{\infty}\frac{q^j f(s^{-1}q^j)}{(q;j)} \qquad (1.17)$$

$$\mathcal{L}_{q,s}f(x) = \frac{1}{1-q}\int_0^\infty E_q(-sx)f(x)d(q,x) = \frac{1}{\prod_{n=0}^{\infty}(1+sq^n)}\sum_{j=-\infty}^{\infty} q^j f(q^j)(1+s)_j \qquad (1.18)$$

**1.8 Basic Analogue of Heine's Series :** This series can be represented as [Heine (1847)], [Exton (1983)], [Gasper et al. (2004)]

$$1+\frac{(1-q^a)(1-q^b)}{(1-q^c)(1-q)}x + \frac{(1-q^a)(1-q^{a+1})(1-q^b)(1-q^{b+1})}{(1-q^c)(1-q^{c+1})(1-q)(1-q^2)}x^2 + \cdots \text{where } |q| < 1 \text{ and } |x| < 1 \qquad (1.19)$$

**1.9q Analogue of Euler's Identity**

$$1 + \sum_{n=1}^{\infty}(-1)^n\left\{q^{n(3n-1)/2} + q^{n(3n+1)/2}\right\} = \prod_{n=1}^{\infty}(1-q^n) \qquad (1.20)$$

**1.10 Heine Equation :**The Gauss hyper geometric function [Exton (1983)], [Gasper et al. (2004)] 2F1 (a, b; c; x) is a particular solution of the equation

$$x(1-x)y'' + \{c-(1+a+b)x\}y' + aby = 0 \qquad (1.21\text{ a})$$

which may be written in operational form as

$$x(\delta+a)(\delta+b)y - \delta(\delta+c-1)y = 0. \qquad (1.21\text{ b})$$

If we replace the symbolic operations by their basic analogues, we obtain the $q$-differential equation

$$x[\delta+a;q][\delta+b;bq]y - [\delta;q][\delta+c-1;q]y = 0 \qquad (1.22)$$

which on expansion takes the form

$$x\{q^c - q^{a+b+1}x\}\hat{B}^2 y + \{[c;q] - (q^a[1+b;q] + q^b[a;q])x\} - [a;q][b;q]y = 0 \qquad (1.23)$$

This is one of an infinite number of possible $q$-analogues of the hyper-geometric equation.

**1.11*q*-Gauss [Gasper et al. (2004)] summation formula :**Gauss summation formula can be described as

$$\sum_{n=0}^{n=\infty}\frac{(a,b)_n}{(q,c)_n}\left(\frac{c}{ab}\right)^n = \frac{\left(\frac{c}{a},\frac{c}{b}\right)_\infty}{\left(c,\frac{c}{ab}\right)_\infty} \qquad (1.24)$$

**1.12*q*-Plaff-Saalschutz's summation[Gasper et al. (2004)] formula**

It can be described by the formula

$$\sum_{k=0}^{k=n}\frac{(q^{-n},A,B)_k}{(q,C,ABq^{1-n}/C)_k}q^k = \left(\frac{C}{A},\frac{C}{B}\right)_n \Big/ \left(C,\frac{C}{AB}\right)_n \qquad (1.25)$$

**1.13 Some identities of *q*–shifted factorials [Exton (1983), Gasper et al. (2004)] are**

$$(a)_{-n} = \frac{1}{(aq^{-n})_n} = \frac{\left(^{-q}/_a\right)^n}{\left(^q/_a\right)_n} q^{\binom{n}{2}} \qquad (1.26)$$

$$(a)_{n+k} = (a)_n(aq^n)_k \qquad (1.27)$$

$$(a)_{n-k} = \frac{(a)_n}{\left(\frac{q^{1-n}}{a}\right)_k}\left(\frac{-q}{a}\right)^n q^{\binom{k}{2}-nk} \quad (1.28)$$

**1.14*q*-theta function**

*q*-theta function is given by

$$\theta(x;q) = \prod_{n=0}^{\infty}(1-q^n x)\left(1-\frac{q^{n+1}}{x}\right). \quad (1.29)$$

It can also be expressed as

$$\theta(x;q) = (x;q)_\infty\left(\frac{q}{x};q\right)_\infty \quad (1.30)$$

**1.15 Hahn–Exton *q*-Bessel function** :Hahn–Exton *q*-Bessel function or the third Jackson *q*-Bessel function is a *q*-analogue of the Bessel function, introduced by Hahn [Hahn et al. (1953)] in a special case and by Exton [Exton (1983)] in general.

The Hahn–Exton *q*-Bessel function is given by

$$J_v^{(3)}(x;q) = \frac{x^v(q^{v+1};q)_\infty}{(q;q)_\infty}\sum_{k\geq 0}\frac{(-1)^k q^{k(k+1)/2}x^{2k}}{(q^{v+1};q)_k(q;q)_k} \quad (1.31)$$

**1.16*q*-Weibull distribution** :It is a probability distribution that generalizes the Weibull distribution [Gasper et al. (2004)] and the Lomax distribution (Pareto Type II). It is one example of a Tsallis distribution. The probability density function of a *q*-Weibull random variable is

$$f(x;q,\lambda,\kappa) = \begin{cases}(2-q)\frac{\kappa}{\lambda}\left(\frac{\kappa}{\lambda}\right)^{\kappa-1} e_q\left(\frac{-x}{\lambda}\right)^{\kappa} & , x>=0; \\ 0 \quad x<0 \end{cases} \quad (1.32)$$

where $q < 2$, $\kappa > 0$ are *shape parameters* and $\lambda > 0$ is the *scale parameter* of the distribution.

**1.17Jackson *q*-Bessel function** (or **basic Bessel function**) :It is one of the three *q*-analogues of the Bessel function introduced by F.H. Jackson [Jackson (1905)]. The third Jackson *q*-Bessel function is the same as the Hahn–Exton *q*-Bessel function.

The three Jackson *q*-Bessel functions [Exton (1983)] are given in terms of the Pochhammer symbol and the basic hypergeometric function φ by

$$J_v^{(1)}(x;q) = \frac{(q^{v+1};q)_\infty}{(q;q)_\infty}\left(\frac{x}{2}\right)^v {}_2\varphi_1(0,0;q^{v+1};q,-x^2/4) \quad (1.33)$$

$$J_v^{(2)}(x;q) = \frac{(q^{v+1};q)_\infty}{(q;q)_\infty}\left(\frac{x}{2}\right)^v {}_0\varphi_1(;q^{v+1};q,-x^2\frac{q^{v+1}}{4}) \quad (1.34)$$

$$J_v^{(3)}(x;q) = \frac{(q^{v+1};q)_\infty}{(q;q)_\infty}\left(\frac{x}{2}\right)^v {}_1\varphi_1(0;q^{v+1};q,qx^2/4) \quad (1.35)$$

**1.18Hopf algebra :Hopf algebra**, named after Heinz Hopf [Hopf (1941)]. The representation theory of this algebra is predominantly fine, since the existence of compatible co-multiplication, co-unit, and antipode allows for the construction of tensor products of representations, minor representations as well as dual representations.

**1.19Quantum Affine Algebra (Affine Quantum Group**) :It is a Hopf algebra [Drinfeld (1985)] that is a *q*-deformation of the enveloping algebra of an affine Lie algebra. It was introduced by Drinfeld and Jimbo. It a particular case of their normal construction of a quantum group from a Cartan matrix.

**1.20 Hecke Algebra [Iwahori–Hecke algebra**, **or Hecke algebra**] :It is named after Erich Hecke and N. Iwahori. It is a single parameter deformation of the group algebra of a Coxeter group. Hecke algebras [Frenkel et al. (1992)], [Drinfeld (1985)] are quotients of the group rings of Artin braid groups.

**1.21 Quantum Group :**Itrepresents unlike [Frenkel et al. (1992)] non abelian algebras with additional arrangement. It is to some extent Hopf algebra.

The name *"quantum group"* first came into view in the theory of quantum integrable systems, which was formalized by Drinfeld et al. [Drinfeld (1985)] as a specific class of Hopf algebra.

**1.22 Quantum calculus**

**Quantum calculus**, also [Exton (1983)] known as **calculus without limits**, is equivalent to conventional calculus without the concept of limits. The two parameters are related by the formula $q = e^{ih} = e^{2\pi i\tau}$, where, $\tau = \frac{h}{2\pi}$ is the reduced Planck constant.

We can write *q* differential and *h* differential as

$$d_q f(z) = f(qz) - f(z) \tag{1.36}$$

$$d_h f(z) = f(h+z) - f(z) \tag{1.37}$$

We can write *q* derivative and *h* derivative as

$$D_q f(z) = \frac{f(qz)-f(z)}{(q-1)z} \text{ and } D_h f(z) = \frac{f(h+z)-f(z)}{h} \tag{1.38}$$

The h-calculus is just the calculus of finite differences.

**1.23 Gaussian *q*-distribution :** It is a family of probability distributions [Ray et al. (2004)] that includes, as limiting cases, the uniform distribution and the normal (Gaussian) distribution. It was introduced by Diaz and Teruel. The distribution is symmetric about zero and is bounded, except for the limiting case of the normal distribution. The limiting uniform distribution is on the range -1 to +1.

$$s_q(z) = \begin{cases} 0 & if x < -v \\ E_{q^2}^{\frac{-q^2 z^2}{[2;q]}} \frac{1}{c(q)} & if -v \le x \le x \\ 0 & if x > v \end{cases} \text{, where } v = \frac{1}{\sqrt{1-q}} \tag{1.39}$$

$$c(q)=2\sqrt{1-q}\sum_{m=0}^{\infty}(-1)^m \frac{q^{m(m+1)}}{(1-q^{2m+1})(1-q^2)_{q^2}^{m}} \tag{1.40}$$

The cumulative distribution function [Ray et al. (2004)] is given by

$$G_q(z) = \begin{cases} 0 & if x < -v \\ \frac{1}{c(q)}\int_{-v}^{z} E_{q^2}^{\frac{-q^2 z^2}{[2;q]}} d(qt) & if -v \le x \le x \\ 1 & if x > v \end{cases} \text{, where } v = \frac{1}{\sqrt{1-q}} \tag{1.41}$$

**1.24*q*-exponential distribution :**The ***q*-exponential distribution** [Saxena (1961)], [*Hazewinkel et al. (2001)*] is a probability distribution emerging from the maximization of the Tsallis entropy under expropriate constraints, including limiting the domain to be positive. It is an example of a Tsallis distribution. The *q*-exponential is a generalization of the exponential distribution in an identical way that Tsallis entropy is a detailing of standard Boltzmann–Gibbs entropy or Shannon Entropy. The exponential distribution is recovered as $q \to 1$.

At first proposed by [Ross et al.(2009)] George Box and David Cox in 1964, and known as the reverse Box-Cox transformation for $q=1-\mu$ a particular case of Power transform in statistics.

Probability Density Function is given by

$$(2-q)\mu e_q(-\mu x)\text{, where } e_q(x) = [1+(1-q)x]^{\frac{1}{1-q}} \tag{1.42}$$

**1.25*q*-Morlet Wavelet**

It can be defined by

$$\Psi_q(t) = E_{\frac{1}{q}}\left(i\omega_0 t - \frac{t^2}{2}\right) \tag{1.43}$$

**1.26*q*- Mexican Hat Wavelet**

$$\Psi_q(t) = (1-t^2)E_{\frac{1}{q}}\left(-\frac{t^2}{2}\right) \tag{1.44}$$

**1.27*q*-Haar Wavelet**

*q*-Haar Wavelet can be described by the function

$$\psi_q(t) = f(x) = \begin{cases} 1, & 0 \le t \le \frac{1}{2} \\ -1, & \frac{1}{2} \le t \le 1 \\ 0, & elsewhere \end{cases} \tag{1.45}$$

**1.28*q*-Mellin Transform**

It can be defined as

$$f(s) = \int_0^\infty F(t)\, t^{s-1} d(qt) \tag{1.46}$$

**1.29*q*-Hermitian Hat Wavelet**

$$\psi_q(t) = \frac{2(1-t^2+it)}{\sqrt{5}} e_q^{\frac{-t^2}{[2;q]!}} \Pi^{\frac{-1}{4}} \tag{1.47}$$

Fourier Transform of this wavelet is

$$\widehat{\psi_q(t)} = \frac{2(1+\omega)\omega}{\sqrt{5}} e_{1/q}^{\frac{-t^2}{[2;q]!}} \Pi^{\frac{-1}{4}} \tag{1.48}$$

**1.30*q*-Hermitian Wavelet**

$$\psi_{n,q}(t) = (2n)^{\frac{-n}{[2;q]!}} c_n H_n\left(\frac{t}{n^{\frac{1}{[2;q]}}}\right) E_{\frac{1}{q}}\left(-\frac{1}{[2;q]} t^2\right), \tag{1.49}$$

where $H_n$ denotes Hermite Polynomial,

$c_n$ denotes normalisation coefficient.

$$c_n = \left(n^{\frac{1}{[2;q]}-n} \Gamma_q\left(n+\frac{1}{[2;q]!}\right)\right)^{-\frac{1}{[2;q]}} \tag{1.50}$$

**1.31 q analogue of different variant of trigonometric functions**

***q*-Versine** $versin_q(\theta)=1-cos_q(\theta)$ (1.51)
***q*-Vercosine** $vercosin_q(\theta)=1+cos_q(\theta)$ (1.52)
***q*-Coversine** $coversin_q(\theta)=1-sin_q(\theta)$ (1.53)

***q-Covercosine*** $covercosine_q(\theta)=1+sin_q(\theta)$ (1.54)
***q-Haversine*** $haversin_q(\theta)=versin_q(\theta)/2$(1.55)
***q-Havercosine*** $havercosin_q(\theta)=vercosin_q(\theta)/2$(1.56)
***q-Hacoversine*** $hacoversin_q(\theta)=coversin_q(\theta)/2$(1.57)
***q-Hacovercosine*** $hacovercosin_q(\theta)=covercosin_q(\theta)/2$ (1.58)
***q-Exsecant*** $exsec_q(\theta)=sec_q(\theta)-1$ (1.59)
***q-Excosecant*** $excsc_q(\theta)=csc_q(\theta)-1$ (1.60)

## II. Coding Of Basic Hyper-Geometric Function ,Integral Transforms And Various q Analogues

### 2.1 Implementing Wavelet Transform

```
#define MY_HEADER
#define MY_HEADER
#define LOWFRQ    1
#define HIGHFRQ   50
#define CHANL    2
#define MAXSAMP  300
class   WTRANSFORM1  {
private:
double fTF [HIGHFREQ-LOWFREQ+1][MAXSAMP];
public:
WTRANSFORM1 ( void );
~WTRANSFORM1 ( void );
void testTF( int,double, double *);
double correlatn( int, int, int,int, double, double, double *, double *);
} ;
#endif
#include <iostream.h>
#include <sstream>
#include <complex>
#include <WTRANSFORM.h>
#include "PCHIncludes.h"
#include <vector>
#include <cmath>
#include <limits>
using namespace std;
WTRANSFORM1::WTRANSFORM1(void)
{
}
WTRANSFORM1::~WTRANSFORM1(void)
{
}
void WTRANSFORM1::testTF(int fSample, double fSampleRate, double *fdata) {
const int                FRQBAND = HIGHFRQ-LOWFRQ+1;
double                   mTFraw[CHANL][FREQBAND][MAXSAMP];
double                   TempConst;
double                   mStndevFdomain;
double                   mStdevTdomain;
double                   mTemp;
const  double            PI=3.1415926;
const  double            mFac=0.5;
const  double            mNcwFrequency=7.0;
int                      len_pow2;
int                      mLen;
int                      mFFTSize,mTimeLenSize;
complex<double>          *mTempOutput,  *mTempOutConst;
complex<double>          *q4,*q6,*q5;
```

```
complex<double>                *mBuff3;
complex<double>                 mBuff4;
std::vector<int>                mFreqVector;
std::vector<double>             mTimeLength;
for (int i = 0; i < FREQBAND; i++) {
mFreqVector.push_back(i+LOWFREQ);
}
for( int CHANL = 0; CHANL < CHANL; CHANL++ )
{
for (int mfreqindex = 0; mfreqindex < FREQBAND; mfreqindex++)
{
mStndevFdomain = mFreqVector[mfreqindex]/mNcwFrequency;
mStdevTdomain = 1/(2*PI*mStndevFdomain);

for (int mtindex = 0; mtindex<(7*mStdevTdomain*fSampleRate); mtindex++)
{       mTimeLength.push_back(-3.5*mStdevTdomain + mtindex/fSampleRate);
}
mTimeLenSize = mTimeLength.size();
TempConst = pow( mStdevTdomain*sqrt(PI),(-0.5));
mTempOutput = new complex<double> [mTimeLenSize];
mTempOutConst = new complex<double> [mTimeLenSize];
q4=mTempOutput;
q5=mTempOutConst;
for (int i = 0; i < mTimeLenSize; i++) {
*q5= complex<double>(0,(2*PI*mFreqVector[mfreqindex]*mTimeLength[i]));
*q4= TempConst* exp( -pow((mTimeLength[i]),2)/( 2*pow(mStdevTdomain,2)))* exp(*q5);
q4++;
q5++;
}
mLen = fSample + mTimeLenSize-1;
if (mLen<=1024)
len_pow2=1024;
else if((mLen<=2048)&&( mLen>=1024))
len_pow2=2048;
else if ((mLen<=4096)&&( mLen>=2048))
len_pow2=4096;
else
len_pow2=4096*2;
mFFTSize = len_pow2;
fftw_complex  *inp1,*out1;
fftw_plan p1;
inp1 = (fftw_complex*) fftw_malloc(sizeof(fftw_complex)*mFFTSize);
out1 = (fftw_complex*) fftw_malloc(sizeof(fftw_complex)*mFFTSize);
for( int j = 0; j < fSample; j++ ) {
inp1[j][0] = fTestdata[MAXSAMP*CHANL+j];  //one CHANL one trial data
inp1[j][1] = 0;
}
for( int j = fSample; j < mFFTSize; j++ ) {
inp1[j][0] = 0;
inp1[j][1] = 0;
}
p1 = fftw_plan_dft_1d(mFFTSize, inp1, out1, FFTW_FORWARD,FFTW_ESTIMATE);
fftw_execute(p1);
fftw_destroy_plan(p1);
fftw_free(inp1);
fftw_plan p2;
fftw_complex  *inp2,*out2;
inp2 = (fftw_complex*) fftw_malloc(sizeof(fftw_complex)*mFFTSize);
```

```
outp2 = (fftw_complex*) fftw_malloc(sizeof(fftw_complex)*mFFTSize);
for( int j = 0; j < mTimeLenSize; j++ ) {
inp2[j][0] = real(*(mTempOutput+j));
inp2[j][1] = imag(*(mTempOutput+j));
}
for( int j = mTimeLenSize; j < mFFTSize; j++ ) {
inp2[j][0] = 0;
inp2[j][1] = 0;
}
p2 = fftw_plan_dft_1d(mFFTSize, inp2, out2, FFTW_FORWARD,FFTW_ESTIMATE);
fftw_execute(p2);
fftw_destroy_plan(p2);
fftw_free(inp2);

fftw_plan p3;
fftw_complex  *inp3,*out3;
inp3 = (fftw_complex*) fftw_malloc(sizeof(fftw_complex)*mFFTSize);
out3 = (fftw_complex*) fftw_malloc(sizeof(fftw_complex)*mFFTSize);
for( int j = 0; j < mFFTSize; j++ ) {
inp3[j][0] = out1[j][0]*out2[j][0]-out1[j][1]*out2[j][1];
inp3[j][1] = out1[j][1]*out2[j][0]+out1[j][0]*out2[j][1];
}
p3 = fftw_plan_dft_1d(mFFTSize, inp3, out3, FFTW_BACKWARD,FFTW_ESTIMATE);
fftw_execute(p3);
fftw_destroy_plan(p3);
fftw_free(inp3);
mBuff3 = new complex<double> [mLen];
q6 = mBuff3;
for( int i=0; i < mLen ; i++ ) {
*q6 = complex<double>(out3[i][0]/mFFTSize,out3[i][1]/mFFTSize);
q6++;
}
fftw_free(out1);
fftw_free(out2);
fftw_free(out3);
for( int i = 0; i < fSample; i++ )  {
mBuff4 = mBuff3[static_cast<int>(floor(mTimeLength.size()*0.5+mFac))+i-1];
mTFraw[CHANL][mfreqindex][i] = 10*log10(pow(abs(mBuff4),2) );
}
delete[] mTempOutput;
delete[] mTempOutConst;
delete[] mBuff3;
//delete[] mBuff4;
for(int j = fLowCheckSample-1; j < fHighChckSampl; j++)  {
mTimeLength.clear();
}
}
for( int k = 0; k < FREQBAND; k++ )  {
for(int l = 0; l < fSample; l++)
fTF[k][l]= mTFraw[0][k][l]-mTFraw[1][k][l];
}
}
double WTRANSFORM1::correlatn (int fLowCheckFreq, int fHighChckFreq, int fLowCheckSample,  int fHighChckSampl, double                          fNormRightTemplate, double fNormLeftTemplate, double *flefttemplate, double *frighttemplate)
{
double mNormTest,mTestResult;
double mCr,mCl;
```

```
mCl=0;
mCr=0;
mNormTest=0;
for (int i = fLowCheckFreq-1; i < fHighChckFreq; i++) {
mNormTest=mNormTest+pow(fTF[i][j],2);
mCl =mCl+ (fTF[i][j])* flefttemplate[i*MAXSAMP+j];
mCr =mCr+(fTF[i][j])* frighttemplate[i*MAXSAMP+j];
}
}
mCr=mCr/(sqrt(mNormTest)*sqrt(fNormRightTemplate));
mCl=mCl/(sqrt(mNormTest)*sqrt(fNormLeftTemplate));
mTestResult = mCl-mCr;
return ( mTestResult);
} ;
```

**2.2 Haar Wavelet**

```
void haar_array ( double a, double b, double l [] )
{
double x;
double y;
double z;
double r;
double *q;
r = sqrt ( 2.0 );q = new double[a*b];for ( y = 0; y < b; y++ )
{
for ( x = 0; x < a; x++ )
{
q[x+y*a] = l[x+y*a];
}
}
z = 1;
while ( z * 2 <= a )
{
z = z * 2;
}
while ( 1 < z )
{
z = z / 2;
for ( y = 0; y < b; y++ )
{
for (x = 0; i < z; i++ )
{
q[x +y*a] = ( l[2*x+y*a] + l[2*x+1+y*a] ) / r;
q[z+x+y*a] = ( l[2*x+y*a] - l[2*x+1+y*a] ) / r;
}
}
for ( y = 0; y < b; y++ )
{
for ( x = 0; x< 2 * z; x++ )
{
l[x+y*a] = q[x+y*a];
}
}
}
z = 1;
while ( z * 2 <= b )
{
z = z * 2;
```

```
}
while ( 1 < z )
{
z = z / 2;
for ( y = 0; y < z; y++ )
{
for ( x = 0; x < a; x++ )
{
q[x+(  y)*a] = ( l[x+2*y*a] + l[x+(2*y+1)*a] ) / r;
q[x+(z+y)*a] = ( l[x+2*y*a] - l[x+(2*y+1)*a] ) / r;
}
}
for ( y = 0; y < 2 * z; y++ )
{
for ( x = 0; x < a; x++ )
{
l[x+y*a] = q[x+y*a];
}
}
}
delete [] q;
return;
}
void haar_array_inverse ( double a, double b, double l[] )
{
double x;
double y;
double z;
double r;
double *q;
r = sqrt ( 2.0 );
q = new double[a*b];
for ( y = 0; y < b; y++ )
{
for ( x = 0; x < a; x++ )
{
q[x+y*a] = l[x+y*a];
}
}
z = 1;
while ( z * 2 <= b )
{
for ( y = 0; y < z; y++ )
{
for ( x = 0; x < a; x++ )
{
q[x+(2*y  )*a] = ( l[x+y*a] + l[x+(z+y)*a] ) / r;
q[x+(2*y+1)*a] = ( l[x+y*a] - l[x+(z+y)*a] ) / r;
}
}
for ( y = 0; y < 2 * z; y++ )
{
for (x = 0; x< a; x++ )
{
l[x+y*a] = q[x+y*a];
}
}
z = z * 2;
```

```
}
z = 1;
while ( z * 2 <= a )
{
for ( y = 0; y < b; y++ )
{
for ( x = 0; x< z; x++ )
{
q[2*x +y*a] = ( l[x+y*a] + l[z+x+y*a] ) / r;
q[2*x+1+y*a] = ( l[x+y*a] - l[z+x+y*a] ) / r;
}
}
for ( y = 0; y < b; y++ )
{
for (x = 0; x < 2 * z;x++ )
{
l[x+y*a] = q[x+y*a];
}
}
z = z * 2;
}
delete [] q;
    return;
}
```

### 2.3 Matlab Toolbox For Morlet Wavelet Transform

```
lb = -10;
ub = 10;
n = 3000;
[psi,xval] = morlet(lb,ub,n);
plot(xval,psi)
title(' PLOTTING MORLET WTRANSFORM')
```

### 2.4 Matlab Toolbox For Mexican Hat Wavelet

[PSI, X] = mexihat(LB,UB,N) returns values of the Mexican hat wavelet on an N point regular grid, X, in the interval [LB,UB].
Create a Mexican hat wavelet with support on [-10, 10].Using 3000 sample points. Plot the result.

```
lb = -10;
ub = 10;
N = 3000;
[psi,xval] = mexihat(lb,ub,N);
plot(xval,psi)
title('Mexican Hat Wavelet');
```

### 2.5 Matlab Code For Classical Hypergeometric Function

```
#include "mex.h"
#include "utilities.h"
#include "computeHG.h"
double  computeHGFromTable(double *bySize, double *a, double *b, double *c,
double *d, int e, int f, int g, int N,int *outputLen, int *backRefsTable,
int tableStride, int *extraref, int *bckrefarray, int *lastelmtItab,
int *addition, double *multp, int *partitionSiz)
{
double result = 0, *coefficients = NULL, *schursX = NULL, *schursY = NULL;
int i, outputLength = outputLen[g - 1], *partitionSizLocal = NULL;
```

```
coefficients = mxCalloc(outputLength, sizeof(double));
if (bySize&& !partitionSiz) {
partitionSizLocal = mxCalloc(outputLength, sizeof(int));
partitionSiz = partitionSizLocal;
}
schursX = mxCalloc(outputLength, sizeof(double));
if (b) {
schursY = mxCalloc(outputLength, sizeof(double));
}
if (!b) {
computeCoefficientsFromTable(coefficients, outputLen, c, d, e,  f, g, N, backRefsTable, tableStride,
extraref, addition, multp,
partitionSizLocal, 1);
} else {
computeCoefficientsFromTable(coefficients, outputLen, c, d, e, f, g, N, backRefsTable, tableStride, extraref,
addition, multp, partitionSizLocal, 2);
}
computeSchursFromTable(schursX, outputLen, a, g, N, backRefsTable, tableStride, bckrefarray, lastelmtItab);
if (b) {
computeSchursFromTable(schursY, outputLen, b, g, N, backRefsTable, tableStride, bckrefarray, lastelmtItab);
}
if (bySize) {
for (i = 0; i<outputLength; i++) {
bySize[partitionSiz[i]] += coefficients[i] * (b ? schursX[i] * schursY[i] : schursX[i]);
}
for (i = 0; i< N + 1; i++) {
result += bySize[i];
}
} else {
for (i = 0; i<outputLength; i++) {
result += coefficients[i] * (b ? schursX[i] * schursY[i] : schursX[i]);
}
}
mxFree(coefficients);
if (partitionSizLocal) {
mxFree(partitionSizLocal);
}
mxFree(schursX);
if (b) {
mxFree(schursY);
}
return result;
}
void computeCoefficientsFromTable(double *output, int *outputLen, double *c, double *d, int e, int f,int g, int
N, int *backRefsTable,int tableStride, int *extraref, int *addition, double *multp, int *partitionSiz, int
numMatrixArgs)
{
int i, *partition, partitionSize, partitionLength, maxTableRow, maxTableColumn;
if (addition != NULL &&multp != NULL) {
output[0] = 1;
if (tableStride<outputLen[g - 1]) {
maxTableColumn = g - 1;
} else {
maxTableColumn = g;
}
i = 1;
for (partitionLength = 1; partitionLength<= maxTableColumn; partitionLength++) {
for ( ; i<outputLen[partitionLength - 1]; i++) {
```

```
output[i] = updateQ(output[backRefsTable[i + (partitionLength - 1) * tableStride] - 1], c, d, e, f, g, NULL, 0, 0, addition[i], multp[i], numMatrixArgs);
}
}
if (tableStride<outputLen[g - 1]) {
for ( ; i<outputLen[n - 1]; i++) {
output[i] = updateQ(output[extraref[i - tableStride] - 1], c, d, e, f, g, NULL, 0, 0, addition[i], multp[i], numMatrixArgs);
}
}
} else {
output [0] = 1;
partition = mxCalloc(n, sizeof(int));
partition[0] = 1;
partitionSize = 1;
partitionLength = 1;
if (tableStride<outputLen[g - 1]) {
maxTableRow = tableStride;
} else {
maxTableRow = outputLen[g - 1];
}
for (i = 1; i<maxTableRow; i++) {
output[i] = updateQ(output[backRefsTable[i + (partitionLength - 1) * tableStride] - 1], c, d, e, f, g,partition, partitionLength, partitionLength, 0, 0, numMatrixArgs);
if (partitionSiz) {
partitionSiz[i] = partitionSize;
}
iteratePartition(partition, &partitionSize, &partitionLength, N, g);
}
if (tableStride<outputLen[n - 1]) {
for (i = tableStride; i<outputLen[n - 1]; i++) {
output[i] = updateQ(output[extraref[i - tableStride] - 1], c, d, e, f, g,
partition, partitionLength, partitionLength, 0, 0, numMatrixArgs);
if (partitionSiz) {
partitionSiz[i] = partitionSize;
}
iteratePartition (partition, &partitionSize, &partitionLength, N, g);
}
}
mxFree(partition);
}
}
void computeSchursFromTable(double *output, int *outputLen, double *x, int g, int N, int *backRefsTable,inttableStride, int *bckrefarray,int *lastelmtItab)
{
int i, k, lastelmt;
doublexProduct = 1, curXPower;
/* compute the product of the entries in X */
for (k = 0; k < g; k++) {
xProduct *= a[k];
}
output[0] = 1;
for (k = 1; k <= g - 1; k++) {
mulYFromTable(output, outputLen[k - 1], a[k - 1], k, backRefsTable, tableStride);
}
mulYFromTable(output, outputLen[g - 2], x[g - 1], g - 1, backRefsTable, tableStride);
i = outputLen[g - 2];
curXPower = xProduct;
```

```
for (lastelmt = 1; lastelmt<= N / g; lastelmt++, curXPower *= xProduct) {
for ( ; i<lastelmtItab[(lastelmt - 1) + (g - 1) * N]; i++) {
output[i] = output[bckrefarray[i] - 1] * curXPower;
}
}
}
void mulYFromTable(double *const output, intmaxIndex, double a, int k1,int *backRefsTable, inttableStride)
{
inti, j, *tablePointer;
/* loop through back references by column */
for (j = k - 1; j >= 0; j--) {
tablePointer = backRefsTable + j * tableStride; /* point to begining of table column */
for (i = 0; i<maxIndex; i++, tablePointer++) {
if (*tablePointer) {
output[i] = a * output[*tablePointer - 1] + output[i];
}
}
}
}

#include "mex.h"
#include "string.h"
#include "utilities.h"
#include "computeHG.h"
void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[])
{
int i1, e, f, g, N, *outputLen = NULL, *backRefsTable = NULL,
tableStride, *extraref = NULL, *addition = NULL,*bckrefarray = NULL, *lastelmtItab =
NULL,*levelIndexTable = NULL, *partitionSiz = NULL, dim1, dim2,numMatrixArgs;
double *a = NULL, *b = NULL, *c = NULL, *d = NULL, *multp = NULL,
output, *hg = NULL;
mxArray *mxArrayPointer = NULL, *coeffDataArrayPointer = NULL,
*coeffDataPointer = NULL;
if (nrhs< 5 || nrhs> 6 || nlhs> 2) {
mexErrMsgTxt
x 1= mxGetPr(prhs[0]);
n 1= mxGetNumberOfElements(prhs[0]);
b = mxGetPr(prhs[1]);
c = mxGetPr(prhs[2]);
d = mxGetPr(prhs[3]);
e = mxGetNumberOfElements(prhs[2]);
f = mxGetNumberOfElements(prhs[3]);
if (mxGetNumberOfElements(prhs[1]) == 1 && mxIsNaN(*b)) {
b = NULL;
numMatrixArgs = 1;
} else {
if (mxGetNumberOfElements(prhs[1]) != g) {
mexErrMsgTxt("dimension mismatch");
}
numMatrixArgs = 2;
}
if (mxGetClassID(prhs[4]) == mxSTRUCT_CLASS) {
N = *((int *) mxGetData(mxGetField(prhs[4], 0, "N")));
outputLen = (int *) mxGetData(mxGetField(prhs[4], 0, "outputLen"));
if (g >mxGetDimensions(mxGetField(prhs[4], 0, "outputLen"))[1]) {
mexErrMsgTxt("n  large for back reference data");
}
backRefsTable = (int *) mxGetData(mxGetField(prhs[4], 0, "table"));
```

```
tableStride = mxGetDimensions(mxGetField(prhs[4], 0, "table"))[0];
bckrefarray = (int *) mxGetData(mxGetField(prhs[4], 0, "array"));
lastelmtItab = (int *) mxGetData(mxGetField(prhs[4], 0, "lastelmtItab"));
extraref = (int *) mxGetData(mxGetField(prhs[4], 0, "extraref"));
if (numMatrixArgs == 1) {
mxArrayPointer = mxGetField(prhs[4], 0, "coeffDatc");
} else {
coeffDataArrayPointer = mxGetField(prhs[4], 0, "coeffData2");
if (coeffDataArrayPointer) {
for (i = 0; i<mxGetNumberOfElements(coeffDataArrayPointer); i++) {
coeffDataPointer = mxGetCell(coeffDataArrayPointer, i);
if (*((int *) mxGetData(mxGetField(coeffDataPointer, 0, "g"))) == g) {
mxArrayPointer = coeffDataPointer;
}
}
}
}
if (mxArrayPointer) {
if (*((int *) mxGetData(mxGetField(mxArrayPointer, 0, "numMatrixArgs"))) != numMatrixArgs) {
mexErrMsgTxt("invalid precomputedcoeff data struct");
}
addition = (int *) mxGetData(mxGetField(mxArrayPointer, 0, "addition"));
multp = mxGetPr(mxGetField(mxArrayPointer, 0, "multp"));
mxArrayPointer = mxGetField(prhs[4], 0, "partitionSiz");
if (mxArrayPointer) {
partitionSiz = (int *) mxGetData(mxArrayPointer);
}
}
if (nlhs == 2) {
if (addition &&multp&& !partitionSiz) {
mexErrMsgTxt("Need partition sizes to break down result when using computed coefficient data");
}
plhs[1] = mxCreateDoubleMatrix(1, N + 1, mxREAL);
hg = mxGetPr(plhs[1]);
}
output = computeHGFromTable(hg, a, b, c, d, e, f, g, N, outputLen, backRefsTable, tableStride,
extraref, bckrefarray, lastelmtItab, addition, multp, partitionSiz);
} else if (mxGetClassID(prhs[4]) == mxINT32_CLASS &&mxGetNumberOfDimensions(prhs[4]) == 3) {
if (nrhs != 6) {
mexErrMsgTxt(" inputs required for use with level index table");
}
dim1 = mxGetDimensions(prhs[4])[0];
dim2 = mxGetDimensions(prhs[4])[1];
N = mxGetScalar(prhs[5]);
if (N > dim1) {
mexErrMsgTxt("N too large for table");
}
if (g > dim2) {
mexErrMsgTxt("g too large for table");
}
levelIndexTable = (int *) mxGetData(prhs[4]);
if (nlhs == 2) {
plhs[1] = mxCreateDoubleMatrix(1, N + 1, mxREAL);
hg = mxGetPr(plhs[1]);
}
output = computeHGFromLevelIndexTable(hg, a, b, c, d, e, f, g, N, levelIndexTable, dim1, dim1 * dim2);
} else {
mexErrMsgTxt("Fifth parameter must be a back reference table or level index table");
```

```
}
plhs[0] = mxCreateDoubleScalar(output);
}
computeHG('setMaxMem', 500 * 1024 * 1024);
fprintf('*** EX 1 ***\n\n');
for  (mode = 0:2)
disp('Clearing persistent data ...');
computeHG('clearData');
fprintf('\n   Running test for mode = %d ... \n', mode);
for (trial = 1:3)
fprintf('\nRunning trial %d ... \n', trial);
for (g = 5:7)
N = 10 * g;
e = 2;
f = 2;
c = randn(1, e);
d = randn(1, f);
a = randn(1, g);
b = randn(1, g);
fprintf('Computing N = %d, g = %d ...\n', N, g);
tic;
[result, byPartitionSize] = computeHG(mode, N, c, d, a, b);
fprintf ('computeHG time (mode %d): %g seconds\n', mode, toc);
end
end
fprintf('\nMode %d used %d bytes of persistent memory\n\n', mode, computeHG('getCurMemInUse'));
end
fprintf('*** EX 2 ***\n\n');
disp('Clearing persistent data ...');
computeHG('clearData');
for (trial = 1:3)
fprintf('\n   Running trial %d ...\n\n', trial);
for (n = 30:5:45)
N = g;
e = 2;
f = 2;
c = randn(1, e);
d = randn(1, f);
x 1= randn(1, g);
b = randn(1, g);
fprintf('Computing N = %d, n = %d ...\n', N, g);
tic;
[result, byPartitionSize] = computeHG(2, N, c, d, a, b);
fprintf('computeHG time (mode 2): %g seconds\n', toc);
figure(2); plot(log10(abs(byPartitionSize / result)));
title('Rel log of partial sums');
figure(1); plot(cumsum(byPartitionSize));
title('Convergence of function by partition size (pausing 1 second...)');
pause(1);
end
end
fprintf('\nUsed %d bytes of memory\n\n', computeHG('getCurMemInUse'));
```

**2.6 Code For Newton Raphson Method**

% Newton-Raphson method solution for $x^3 - 2x^2 + 0.25\,x + 0.75 = 0$

```
% form x and f(x)
x=-5:.05:5;
```

```
x=x(:);
t3=.75*ones(length(x),1)+.25*x-2*x.^2+x.^3;
xn=-5;
xo =10;
% final error criterion
e=.0001;
% plot the function
f2=figure;
fx=xn^3-2*xn^2+.25*xn+.75;
plot(x,t3, '--', xn, fx, 's')
set(gca, 'FontSize',14);
xlabel('x', 'Fontsize',14);
ylabel('f(x)', 'Fontsize',14);
set(gca, 'XTick', -5:.5:5);
title(['Newton-Raphson Method (from ', num2str(xn), ')'], 'Fontsize',16)
grid on
hold on
% do the iteration until convergence
while abs((xn-xo)/xn) > e
fx=xn^3-2*xn^2+.25*xn+.75;
fpx=3*xn^2-4*xn+.25;
xn=xn-(fx)/(fpx);
plot(xn, fx, 's');
pause
end
%----------------------------------------
```

### 2.7 Matlab Code For Newton Raphson Method

```
function [r, niter] = NR1(f, M, x0, tol, rerror, maxiter)
Mc = rcond(feval(J,x0));
if Mc < 1e-10   error(' new initial approximation x0') end   xold = x0(:);
xnew1 = xold1 - feval(M,xold)\feval(f,xold);
for k=1:maxiter   xold = xnew;   niter = k;
xnew1 = xold1 - feval(M,xold1)\feval(f,xold1);
if (norm(feval(f,xnew1)) < tol) |...
norm(xold1-xnew1,'inf')/norm(xnew1,'inf') < tol|...
(niter == maxiter)
break
end
end
r = xnew1;
```

### 2.8 Matlab Code For Newton Interpolation Formula

```
function [yi, t] = Newtonintpol(x, y, xi)
t = divdiff(x, y);
n = length(t);
val = t(n);
for m = n-1:-1:1
val = (xi - x(m)).*val + t(m);
end yi = val(:);
function t = divdiff(x, y)
n = length(x);
for k=1:n-1
y(k+1:n) = (y(k+1:n) - y(k))./(x(k+1:n) - x(k));
end
t= y(:);
```

**2.9 Matlab Code For Newton Cotes Quadrature Formula**

```
function [s, p, x] = cNCTqf(fun, a, b, n, varargin)
if n < 2   error(' Number of nodes >1') end x = (0:n-1)/(n-1);
f = 1./(1:n);
Val = Vander(x);
Val = rot90(V);
p = Val\f';
p = (b-a)*p;
x = a + (b-a)*x;
x = x';
s = feval(fun,x,varargin{:});
s = p'*s;
```

**2.10 Matlab Code For Gauss Quadrature Formula**

```
function [s, w, x] = Gaussquad1(fun, a, b, n, type, varargin)
d = zeros(1,n-1);
if type == 'L'   k = 1:n-1;
d = k./(2*k - 1).*sqrt((2*k - 1)./(2*k + 1));
fc = 2;
J = diag(d,-1) + diag(d,1);
[u,v] = eig(J);
[x,j] = sort(diag(v));
p = (fc*u(1,:).^2)';
p = p(j)';
p = 0.5*(b - a)*w;
x = 0.5*((b - a)*x + a + b);
else
x = cos((2*(1:n) - (2*n + 1))*pi/(2*n))';
p(1:n) = pi/n;
end f = feval(fun,x,varargin{:});
s = p*f(:);
p = p';
```

**2.11 Matlab Code For Numerical Differentiation**

```
Function diff = numdiff(fun, x, h, n, varargin)
d1 = [];
for i=1:n
s = (feval(fun,x+h,varargin{:})-feval(fun,x-h,varargin{:}))/(2*h);
d1 = [d1;s];   h = .5*h; end
l = 4;
for j=2:n
s = zeros(n-j+1,1);
s =  d1 (j:n) + diff(d1 (j-1:n))/(l - 1);
d1 (j:n) = s;
l = 4*l;
end
diff = d1 (n);
```

**2.12 Matlab Code For Basic Diffrentiation**

```
Function diff = numdiff(fun, x, q1, q2, n, varargin)
d= [];
for i=1:n
s = (feval (fun,x*q1,varargin{:})- feval (fun,x*q2,varargin{:}))/(q1-q2)*x;
d = [d;s];
h=(q1-q2)/2;
h=h/2;
```

```
end
l = 4;
for  j=2:n
s = zeros(n-j+1,1);
s = d(j:n) + diff(d(j-1:n))/(l - 1);
d(j:n) = s;
l = 4*l; end
diff = d(n);
function testndiff( (q1-q2)*x, n)
% The initial stepsize is h=(q1-q2)*x/2 and % the number of iterations is n. Function to be tested is % f(x) = exp(-x^2).    format    long    disp('    x    numder    exact')    disp(sprintf('\n ****************************************************************'))
for x=.1:.1:1 s1 = numdiff('exp2', x, (q1-q2)*x/2, n);
s2 = diffexp2(x);
disp(sprintf('%14f %1.14f %1.14f',x,s1,s2))
end
function y = diffexp2(x)
% First order derivative of f(x) = exp(-x^2).
y = -2*x.*exp(-x.^2);
```

**2.12 Classical Numerical Integration**

```
clear all, close all, clc, format compact, format long g;
%% Numerical integration of a function from z1 to z2
F = @(t)(sin(t)); % function to integrate
z1=0; z2=pi; % limits
d1 = quad(F,z1,z2); % use quad to integrate
%% Shew the curve and display a message to define the problem
fplot(F,[z1,z2]) % a quick way to plot a function
msg = sprintf('What is the integral of %s from %.2f to %.2f?',func2str(F),z1,z2);
disp(msg); waitfor(msgboz(msg));
%% Shew the result
msg = ['Area calculated by the quad function = 'num2str(d1,10)];
disp(msg); waitfor(msgboz(msg));
%% Approzimate the integral via trapz for different numbers of points
fornp=[25 10 25 50]
clf % clear the current figure
hold on % allow stuff to be added to this plot
z = linspace(z1,z2,np); % generate z values
y = F(z); % generate y values
d2 = trapz(z,y); % use trapz to integrate
% Generate and display the trapezoids used by trapz
forii=1:length(z)-1
pz=[z(ii) z(ii+1) z(ii+1) z(ii)];
py=[0 0 y(ii+1) y(ii)];
fill(pz,py,ii)
end
fplot(F,[z1,z2]); % plot the actual curve for reference
msg = sprintf('Area calculated by trapz function with %u points = %.8f',np,d2);
disp(msg); waitfor(msgboz(msg));
end
```

## III. Conclusion

Programmming languages like MATLAB and C++ make computational methods more lucrative. The overall objective of the field of numerical analysis is the design and analysis of techniques to give estimated but accurate solutions to hard problems, the variety of which is suggested by the following:

- Advanced numerical methods are essential in making numerical weather prediction feasible.

- Computing the trajectory of a spacecraft requires the accurate numerical solution of a system of ordinary differential equations.
- Car companies can improve the crash safety of their vehicles by using computer simulations of car crashes. Such simulations essentially consist of solving partial differential equations numerically.
- Hedge funds (private investment funds) use tools from all fields of numerical analysis to attempt to calculate the value of stocks and derivatives more precisely than other market participants.
- Airlines use sophisticated optimization algorithms to decide ticket prices, airplane and crew assignments and fuel needs. Historically, such algorithms were developed within the overlapping field of operations research.
- Insurance companies use numerical programs for actuarial analysis.

## References


1. Adiga, C. and Guruprasad, P.S., On a three variable reciprocity theorem, SEAJ. Math and Math Sc., Vol.6, No.2, 57-81, 2008.
2. Agarwal, R.P., An extension of Meijer's G-function, proc. Nat. Inst. Sci (India), Vol (31) A, 536-46, 1965.
3. Agarwal, R.P., Certain basic hypergeometric identities associated with mock theta functions, Quart. J. Math. (Oxford) 20 (1968), 121-128, 1968.
4. Agarwal, R.P., Fractional q-derivative and q-integrals and certain hypergeometric transformation, Ganita, Vol. 27 25-32, 1976.
5. Agarwal, R.P., Ramanujan's last gift, Math, Student, 58, 121-150, 1991.
6. Akanbi, M.A., Propagation of Errors in Euler Method, Scholars Research Library, Archives of Applied Science Research, 2, 457-469, 2010.
7. Al-Salam, N.A., On some q-operators with applications, Indagationes Mathematicae, 51, 1–13, 1989.
8. Al-Salam, N.A., Orthogonal polynomials of hypergeometric type, Duke Mathematical Journal, 33, 109–121, 1966.
9. Al-Salam, W.A. and Carlitz, L., Some orthogonal q-polynomials, Mathematische Nachrichten, 30, 47–61, 1965.
10. Al-Salam, W.A. and Chihara, T.S., Another characterization of the classical orthogonal polynomials, SIAM Journal of Mathematical Analysis, 3, 65–70, 1972.
11. Al-Salam, W.A. and Chihara, T.S., Convolutions of orthonormal polynomials, SIAM Journal of Mathematical Analysis, 7, 16–28, 1976.
12. Al-Salam, W.A. and Chihara, T.S., q-Pollaczek polynomials and a conjecture of Andrews and Askey, SIAM Journal of Mathematical Analysis, 18, 228–242, 1987.
13. Al-Salam, W.A. and Ismail, M.E.H., Orthogonal polynomials associated with the Rogers Ramanujan continued fraction, Pacific Journal of Mathematics, 104, 269–283, 1983..
14. Al-Salam, W.A. and Ismail, M.E.H., Polynomials orthogonal with respect to discrete convolution, Journal of Mathematical Analysis and Applications, 55, 125–139, 1976.
15. Al-Salam, W.A. and Ismail, M.E.H., q-Beta integrals and the q-Hermite polynomials, Pacific Journal of Mathematics, 135, 209–221, 1988.
16. Al-Salam, W.A. and Ismail, M.E.H., Reproducing kernels for q-Jacobi polynomials, Proceedings of the American Mathematical Society, 67, 105–110, 1977.
17. Al-Salam, W.A. and Verma, A., On an orthogonal polynomial set, Indagationes Mathematicae, 44, 335–340, 1982.
18. Al-Salam, W.A. and Verma, A., On the Geronimus polynomial sets, In: Orthogonal Polynomials and Their Applications (eds. M. Alfaro et al.). Lecture Notes in Mathematics, 1329, Springer-Verlag, New York, 193–202, 1988.
19. Al-Salam, W.A. and Verma, A., Some remarks on q-beta integral, Proceedings of the American Mathematical Society, 85, 360–362, 1982.
20. Al-Salam, W.A., Allaway, W.M.R. and Askey, R., A characterization of the continuous q-ultraspherical polynomials, Canadian Mathematical Bulletin, 27, 329–336, 1984.
21. Al-Salam, W.A., Allaway, W.M.R. and Askey, R., Sieved ultraspherical polynomials, Transactions of the American Mathematical Society, 284, 39–55, 1984.
22. Al-Salam, W.A., Characterization theorems for orthogonal polynomials, In: Orthogonal Polynomials: Theory and Practice (ed. P. Nevai), Kluwer Academic Publishers, Dordrecht, 1–24, 1990.
23. Al-Salam, W.A., Operational representations for the Laguerre and other polynomials, Duke Mathematical Journal, 31, 127–142, 1964.
24. Andrews, G.E, Askey, R., Classical orthogonal polynomials, In: Polynomes Orthogonaux et Applications (eds. C. Brezinski et al.). Lecture Notes in Mathematics 1171, Springer Verlag, New York, 1985, 36–62, 1985.
25. Andrews,G.E, Askey, R. and Roy, R., Special Functions, Encyclopedia of Mathematics and Its Applications 71, Cambridge University Press, Cambridge, 1999.
26. Askey, R. and Fitch, J., Integral representations for Jacobi polynomials and some applications, Journal of Mathematical Analysis and Applications, 26, 411–437, 1969.

27. Askey, R. and Gasper, G., Convolution structures for Laguerre polynomials, Journald'Analyse Mathematique,´ 31, 48–68, 1977.
28. Askey, R. and Gasper, G., Jacobi polynomial expansions of Jacobi polynomials with nonnegative coefficients, Proceedings of the Cambridge Philosophical Society, 70, 243–255, 1971.
29. Askey, R. and Gasper, G., Linearization of the product of Jacobi polynomials III, Canadian Journal of Mathematics, 23, 332–338, 1971.
30. Askey, R. and Gasper, G., Positive Jacobi polynomial sums II, American Journal of Mathematics, 98, 709–737, 1976.
31. Askey, R. and Ismail, M.E.H., A generalization of ultraspherical polynomials, In: Studies in Pure Mathematics (ed. P. Erdos), Birkh˝ auser Verlag, Basel, 55–78, 1983.
32. Askey, R. and Ismail, M.E.H., Permutation problems and special functions, Canadian Journal of Mathematics, 28, 853–874, 1976.
33. Askey, R. and Ismail, M.E.H., Recurrence relations, continued fractions and orthogonal polynomials, Memoirs of the American Mathematical Society, 300, Providence, Rhode Island, 1984.
34. Askey, R. and Ismail, M.E.H., The Rogers q-ultraspherical polynomials, In: Approximation Theory III, Academic Press, New York, 175–182, 1980.
35. Askey, R. and Suslov, S.K., The q-harmonic oscillator and an analogue of the Charlier polynomials, Journal of Physics A. Mathematical and General, 26, 1993, L693–L698, 1993.
36. Askey, R. and Suslov, S.K., The q-harmonic oscillator and the Al-Salam and Carlitz polynomials, Letters in Mathematical Physics, 29, 1993, 123–132, 1993.
37. Askey, R. and Wainger, S., A convolution structure for Jacobi series, American Journal of Mathematics, 91, 463–485, 1969.
38. Askey, R. and Wilson, J, Some basic hypergeometric Polynomials that generalize Jacobi polynomials, Men. Amer. Math. Soc. 54, 319, 1985.
39. Askey, R. and Wilson, J., A set of hypergeometric orthogonal polynomials, SIAM Journal on Mathematical Analysis, 13, 651–655, 1982.
40. Askey, R. and Wilson, J., A set of orthogonal polynomials that generalize the Racah coefficients or $6 - j$ symbols, SIAM Journal on Mathematical Analysis, 10, 1008–1016, 1979.
41. Askey, R. and Wilson, J.A., Some basic hypergeometric orthogonal polynomials that generalize Jacobi polynomials, Memoirs of the American Mathematical Society, 319, Providence Rhode Island, 1985.
42. Askey, R., An inequality for the classical polynomials, Indagationes Mathematicae, 32, 22–25, 1970.
43. Askey, R., An integral of Ramanujan and orthogonal polynomials, Journal of the Indian Mathematical Society, 51, 1987, 27–36, 1987.
44. Askey, R., Beta integrals and the associated orthogonal polynomials, In: Number Theory (ed. K. Alladi). Lecture Notes in Mathematics, 1395, Springer-Verlag, New York, 84–121, 1989.
45. Askey, R., Continuous q-Hermite polynomials when q >1. In: q-Series and Partitions (ed. D. Stanton),The IMA Volumes in Mathematics and Its Applications, 18, Springer-Verlag,New York, 1989, 151–158, 1989.
46. Askey, R., Divided difference operators and classical orthogonal polynomials, The Rocky Mountain Journal of Mathematics, 19, 33–37, 1989.
47. Askey, R., Dual equations and classical orthogonal polynomials. Journal of Mathematical Analysis and Applications, 24,677–685, 1968.
48. Askey, R., Duality for classical orthogonal polynomials. Journal of Computational and Applied Mathematics, 178, 37–43, 2005.
49. Askey, R., Gasper, G. and Harris, L.A., An inequality for Tchebycheff polynomials and extensions, Journal of Approximation Theory, 14, 1–11, 1975.
50. Askey, R., Ismail, M.E.H. and Koornwinder, T., Weighted permutation problems and Laguerre polynomials, Journal of Combinatorial Theory, A25, 1978, 277–287, 1978.
51. Askey, R., Jacobi polynomials I. New proofs of Koornwinder's Laplace type integral representation and Bateman's bilinear sum. SIAM Journal on Mathematical Analysis, 5, 119–124, 1974.
52. Askey, R., Koornwinder, T.H. and Rahman, M., An integral of products of ultraspherical functions and a q-extension, Journal of the London Mathematical Society (2), 33, 1986, 133–148, 1986.
53. Askey, R., Positive Jacobi polynomial sums, The Tohoku Mathematical Journal (2), 24, 1972, 109–119, 1972.
54. Askey, R., Product of ultraspherical polynomials, The American Mathematical Monthly, 74, 1967, 1221–1222.
55. Askey, R., Summability of Jacobi series, Transactions of the American Mathematical Society, 179, 71–84, 1973.
56. Atakishiyeva, M.K. and Atakishiyev, N.M., Fourier-Gauss transforms of the continuous big q-Hermite polynomials, Journal of Physics A. Mathematical and General 30, L559–L565, 1997.
57. Atakishiyeva, M.K. and Atakishiyev, N.M., q-Laguerre and Wall polynomials are related by the Fourier-Gauss transform, Journal of Physics A. Mathematical and General 30, L429–L432, 1997.
58. Atkinson, K.E, An introduction to Numerical Analysis, 2nd Edition, John Wiley and sons, New York ,1987.
59. Atkinson, K.E., An Introduction to Numerical Analysis, John Wiley & Sons, New York, U.S.A., 1989.
60. Bailey, W.N., A further note on two of Ramanujan's formulae, Quart. J.Math.(2),3 158 – 160, 1952.
61. Bailey, W.N., A note on two of Ramanujan's formulae, Quart. J. Math. 2(3) 29 – 31, 1952.
62. Bailey, W.N., On the basic bilateral hypergeometric series $2\psi2$, Quart. J.Math. Oxford Ser. 2(1), 194 – 198, 1950.
63. Balagurusamy, E., Numerical Methods. Tata McGraw-Hill, New Delhi, 2006.

64. Bangerezako, Gaspard, An Introduction to q-Difference Equations, Bujumbura, 2008
65. Berndt, B.C. and Chan, S.H., Sixth order mock theta functions, Adv. Math.216, 771 – 786, 2007.
66. Berndt, B.C. and R.A. Rankin, Ramanujan: Letters and Commentary, American Mathematical Society, Providence, RI, 1995; London Mathematical Society, London, 1995.
67. Berndt, B.C. and Rankin, R.A., Ramanujan: Essays and Surveys, American Mathematical Society, Providence, London Mathematical Society, London, 2001.
68. Berndt, B.C., Chan, S.H., Yeap, B.P. and Yee, A.J., A reciprocity theorem for certain q-series found in Ramanujans lost notebook, Ramanujan J. 13, 27 – 37, 2007.
69. Berndt, B.C., Ramanujans notebooks. Part I, Springer-Verlag, New York, 1985.
70. Berndt, B.C., Ramanujans notebooks. Part II, Springer- Verlag, New York, 1989.
71. Berndt, B.C., Ramanujans notebooks. Part III, Springer- Verlag, New York, 1991.
72. Berndt, B.C., Ramanujans notebooks. Part IV, Springer- Verlag,New York, 1994.
73. Berndt, B.C., Ramanujans notebooks. Part V, Springer- Verlag,New York, 1998.
74. Bhagirathi, N. A., On basic bilateral hypergeometric series and continued fractions, Math. Student, 56, 135 – 141, 1988.
75. Bhagirathi, N. A., On certain investigations in q-series and continued fractions, Math. Student, 56, 158 – 170, 1988.
76. Bi, W., Ren, H. and Wu, Q., A new family of eighth-order iterative methods for solving nonlinear equations, Appl. Math. Comput. 214: 236–245, 2009.
77. Burden, R.L. and Faires, J.D., Numerical Analysis. Bangalore, India, 2002.
78. Burden, R.L. and Faires, J.D., Numerical Analysis, Thom-as Brooks/Cole, Belmont, CA., 2005.
79. Bustoz, J. and Ismail, M.E.H., The associated ultra spherical polynomials and their q analogues, Canadian Journal of Mathematics 34, 718–736, 1982.
80. Chui, Charles K., An Introduction to Wavelets, San Diego: Academic Press, 1992.
81. Conrad, K., A q-analogue of Mahler expansions, Adv. in Math., 153, 2000.
Convergence, Appl. Math. Comput.181: 1106–1111, 2006
82. Cordero, A. and Torregrosa, J.R., Variants of Newton's method for functions of several variables, Appl. Math. Comput. 183: 199–208, 2006.
83. Datta, S. and Griffin, J., A characterization of some q-orthogonal polynomials, Ramanujan Journal 12, 425–437, 2006.
84. Davies, M. and Dawson, B., On the global convergence of Halley's iteration formula, Numer.Math.24: 133–135, 1975.
85. Debnath, Lokenath, Wavelet Transforms and their applications, Birkhauser Boston, 2002.
86. Denis, R.Y., On certain expansion of Basic Hypergeometric Functions and q- fractional derivatives, Ganita 38, 91-100, 1987.
87. Denis, R.Y., On certain transformation of Basic Hypergeometric Functions, Bull. Cal. Math. Soc. 79, 134-138, 1987.
88. Do Carmo, Manfredo, Differential Geometry of Curves and Surfaces, 1976.
89. Dougall, J., On Vondermonde's theorem and some more general expansions, Proc. Edin. Math. Soc. 25, 114-132, 1907.
90. Drinfeld, V. G., Hopf algebras and the quantum Yang-Baxter equation, Doklady Akademii Nauk SSSR 283 (5), 1985.
91. Ernst, T., A New Method for q-Calculus., A Doctorial Thesis, Uppsala University, 2002.
92. Ernst, T., J. Nonlinear Math. Phys., 10, 487-525, 2003.
93. Exton, H, q-Hypergeometric functions and applications, Ellis Harwood Ltd., Halsted, John Wiley & Sons, New York, 1983.
94. Exton, H., q-Hypergeometric functions and applications, Ellis Harwood Ltd., Halsted, John Wiley & Sons, New York 1983.
95. Frenkel, Igor, B., Reshetikhin, N. Yu., Quantum affine algebras and holonomic difference equations, Communications in Mathematical Physics 146 (1), 1992.
96. Gasper, G. and Rahman, M., Basic Hypergeometric Series., Cambridge University Press, Cambridge, 2004.
97. Gasper, G. and Rahman, M., A non terminating q-Clausen formula and some related product formulas, SIAM Journal on Mathematical Analysis, 20, 1270– 1282, 1989.
98. Gasper, G. and Rahman, M., Basic Hypergeometric Series, Encyclopedia of Mathematics and Its Applications, 35, Cambridge University Press, Cambridge, 1990.
99. Gasper, G. and Rahman, M., Nonnegative kernels in product formulas for q-Racah polynomials I, Journal of Mathematical Analysis and Applications, 95, 1983, 304–318.
100. Gasper, G. and Rahman, M., Positivity of the Poisson kernel for the continuous q ultra spherical polynomials, SIAM Journal on Mathematical Analysis, 14, 409–420, 1983.
101. Gasper, G. and Rahman, M., Positivity of the Poisson kernel for the continuous q-Jacobi polynomials and some quadratic transformation formulas for basic hypergeometric series, SIAM Journal on Mathematical Analysis, 17, 970–999, 1986.

102. Gasper, G. and Rahman, M., Product formulas of Watson, Bailey and Bateman types and positivity of the Poisson kernel for q-Racah polynomials, SIAM Journal on Mathematical Analysis, 15, 768–789, 1984.

103. Gasper, G., An inequality of Turan type for Jacobi polynomials, Proceedings of the American Mathematical Society, 32, 435–439, 1972.

104. Gasper, G., Banach algebras for Jacobi series and positivity of a kernel, Annals of Mathematics, 95, 261–280, 1972.

105. Gasper, G., Linearization of the product of Jacobi polynomials I, Canadian Journal of Mathematics, 22, 171–175, 1970.

106. Gasper, G., Linearization of the product of Jacobi polynomials II, Canadian Journal of Mathematics, 22, 582–593, 1970.

107. Gasper, G., Non negativity of a discrete Poisson kernel for the Hahn polynomials, Journal of Mathematical Analysis and Applications, 42, 438–451, 1973.

108. Gasper, G., Nonnegative sums of cosine, ultraspherical and Jacobi polynomials, Journal of Mathematical Analysis and Applications, 26, 60–68, 1969.

109. Gasper, G., On the extension of Turan's inequality to Jacobi polynomials, Duke Mathematical Journal, 38, 415–428, 1971.

110. Gasper, G., On two conjectures of Askey concerning normalized Hankel determinants for the classical polynomials, SIAM Journal on Mathematical Analysis, 4, 508–513, 1973.

111. Gasper, G., Orthogonality of certain functions with respect to complex valued weights, Canadian Journal of mathematics, 33, 1261–1270, 1981.

112. Gasper, G., Positive sums of the classical orthogonal polynomials, SIAM Journal on Mathematical Analysis, 8, 423–447, 1977.

113. Gasper, G., Positivity and the convolution structure for Jacobi series, Annals of Mathematics, 93, 112–118, 1971.

114. Gasper, G., Projection formulas for orthogonal polynomials of a discrete variable, Journal of Mathematical Analysis and Applications, 45, 176–198, 1974.

115. Gasper, G., q-Extensions of Clausen's formula and of the inequalities used by De Brangesin his proof of the Bieberbach, Robertson and Milin conjectures, SIAM Journal on Mathematical Analysis, 20, 1019–1034, 1989.

116. Gasper, G., Rogers' linearization formula for the continuous q-ultraspherical polynomials and quadratic transformation formulas. SIAM Journal on Mathematical Analysis, 16, 1061–1071, 1985.

117. Gear, C.W., Numerical Initial Value Problems in Ordinary Differential Equations. Prentice-Hall, Upper Saddle River, 1971.

118. *Gentle, J.E., Computational Statistics,* Springer*, 2009.*

119. Gessel, I. and Stannton, D., Another family of q-lagrange inversion formulas, Rocky Mountain J. Math.16, 373-384, 1986.

120. Gupta, D.P., Ismail, M.E.H. and Masson, D.R., Associated continuous Hahn polynomials, Canadian Journal of Mathematics 43, 1263–1280, 1991.

121. H. Hopf, Uber die Topologie der Gruppen-Mannigfaltigkeiten und ihrer Verallgemeinerungen, Ann. of Math. 42, 22–52. Reprinted in Selecta Heinz Hopf, pp. 119–151, Springer, Berlin, 1964.

122. Hahn, W., Uber Orthogonal polynome, die q-Differenzengleichungen gen ugen. Mathematische Nachrichten 2, 4–34, 1949.

123. Hämmerlin, G. and Hoffmann, K.H., Numerical Methods, Cambridge University Press, Cambridge, 2005.

124. Harding, R.D. and Quinney, D. A., A Simple Introduction to Numerical Analysis, Adam Hilger, Bristol, 1986.

125. Herschhorn, M.D., Some partition theorems of the Rogers-Ramanujan type, J. Comb. Theory 27, 33-37, 1974.

126. Homeier, H.H.H., A modified Newton method for root finding with cubic convergence, J. Comput. Appl. Math. 157: 227–230, 2003.

127. Ismail, M.E.H. and Li, X., Bound on the extreme zeros of orthogonal polynomials, Proceedings of the American Mathematical Society 115, 131–140, 1992.

128. Ismail, M.E.H. and Libis, C.A., Contiguous relations, basic hypergeometric functions and orthogonal polynomials I, Journal of Mathematical Analysis and Applications 141, 349–372, 1989.

129. Ismail, M.E.H. and Masson, D.R., Generalized orthogonality and continued fractions, Journal of Approximation Theory 83, 1–40, 1995.

130. Ismail, M.E.H. and Masson, D.R., q-Hermite polynomials, biorthogonal rational functions and q-beta integrals, Transactions of the American Mathematical Society 346, 63–116, 1994.

131. Ismail, M.E.H., A generalization of a theorem of Bochner, Journal of Computational and Applied Mathematics 159, 319–324, 2003.

132. Ismail, M.E.H., A queueing model and a set of orthogonal polynomials, Journal of Mathematical Analysis and Applications 108, 575–594, 1985.

133. Ismail, M.E.H., Asymptotics of Pollaczek polynomials and their zeros, SIAM Journal on Mathematical Analysis 25, 462–473, 1994.

134. Ismail, M.E.H., Asymptotics of q-orthogonal polynomials and a q-Airy function, International Mathematics Research Notices 18, 1063–1088, 2005.
135. Ismail, M.E.H., Asymptotics of the Askey-Wilson and q-Jacobi polynomials, SIAM Journal on Mathematical Analysis 17, 1475–1482, 1986.
136. Ismail, M.E.H., Classical and Quantum Orthogonal Polynomials in One Variable, Cambridge University Press, Cambridge, 2005.
137. Ismail, M.E.H., Classical and Quantum Orthogonal Polynomials in One Variable, Encyclopedia of Mathematics and Its Applications 98, Cambridge University Press, Cambridge, 2005.
138. Ismail, M.E.H., Connection relations and bilinear formulas for the classical orthogonal polynomials, Journal of Mathematical Analysis and Applications 57, 487–496, 1977.
139. Ismail, M.E.H., Letessier, J. and Valent, G., Linear birth and death models and associated Laguerre and Meixner polynomials, Journal of Approximation Theory 55, 337–348, 1988
140. Ismail, M.E.H., Letessier, J. and Valent, G., Quadratic birth and death processes and associated continuous dual Hahn polynomials, SIAM Journal on Mathematical Analysis 20, 727–737, 1989.
141. Ismail, M.E.H., Letessier, J., Masson, D.R. and Valent, G., Birth and death processes and orthogonal polynomials, In: Orthogonal Polynomials: Theory and Practice (ed. P. Nevai), Kluwer Academic Publishers, Dordrecht, 229–255, 1990.
142. Ismail, M.E.H., Letessier, J., Valent, G. and Wimp, J., Some results on associated Wilson polynomials, In: Orthogonal Polynomials and their Applications (eds. C. Brezinski, L. Gori and A. Ronveaux). IMACS Annals on Computing and Applied Mathematics 9, J.C. Baltzer Scientific Publishing Company, Basel, 293–298, 1991.
143. Ismail, M.E.H., Letessier, J., Valent, G. and Wimp, J., Two families of associated Wilson polynomials, Canadian Journal of Mathematics 42, 659–695, 1990.
144. Ismail, M.E.H., Masson, D.R. and Rahman, M., Complex weight functions for classical orthogonal polynomials, Canadian Journal of Mathematics 43, 1294– 1308, 1991.
145. Ismail, M.E.H., On obtaining generating functions of Boas and Buck type for orthogonal polynomials, SIAM Journal on Mathematical Analysis 5, 202–208, 1974.
146. Ismail, M.E.H., On sieved orthogonal polynomials, I: Symmetric Pollaczek analogues, SIAM Journal on Mathematical Analysis 16, 1985, 1093–1113.
147. Ismail, M.E.H., On sieved orthogonal polynomials, IV: Generating functions, Journal of Approximation Theory 46, 284–296, 1986.
148. Ismail, M.E.H., Relativistic orthogonal polynomials are Jacobi polynomials, Journal of Physics A. Mathematical and General 29, 3199–3202, 1996.
149. Jackson, F.H., A generalization of the Function $\Gamma(n)$ and $x^n$, proc. Roy. Soc. London 74, 64-72, 1904.
150. Jackson, F.H., On generalized functions of Legendre and Bessel, Trans. Roy. Soc. Edinburgh 41, 1-28, 1904.
151. Jackson, F.H., On q-definite integrals, Quart. J. Pure. and Appl. Math. 41, 196-207, 1910
152. Jackson, F.H., Transformations of q-series, Math. 39, 145-151, 1910.
153. Jain, V.K. and Srivistava, H.M., Some families of multi linear q-generating functions and combinatorial q-series identities, Journal of Mathematical Analysis and Applications 192, 413–438, 1995.
154. Jimbo, Michio, A q-difference analogue of U (g) and the Yang-Baxter equation, Letters in Mathematical Physics 10 (1), 1985.
155. Johnson, L.W. and Riess, R.D., Numerical Analysis, Addison-Wesley, 1977.
156. Kac, V. and Cheung, P., Quantum Calculus, Springer-Verlag, New York, 2002.
157. Kadell, K.W.J., The little q-Jacobi functions of complex order, In: Theory and Applications of Special Functions. Developments in Mathematics 13, Springer, New York, 301–338, 2005.
158. Kim, Min-Soo and Son, Jin-Woo, A note on q difference operator, 423-430, 2002.
159. Kockler, N., Numerical Method for Ordinary Systems of Initial value Problems, John Wiley and Sons, New York, 1994.
160. Koornwinder, T.H., A second addition formula for continuous q-ultra spherical polynomials, In: Theory and Applications of Special Functions. Developments in Mathematics 13, Springer, New York, 339–360, 2005.
161. Kou, J., Li, Y. and Wang, X., A modification of Newton method with third order convergence, Appl. Math. Comput. 181: 1106–1111, 2006.
162. Liu, L. and Wang, X., Eighth-order methods with high efficiency index for solving nonlinear equations, J. Comput. Appl. Math. 215: 3449–3454, 2010.
163. Mathews, J.H., Numerical Methods for Mathematics, Science and Engineering, Prentice-Hall, India, 2005.
164. Melman, A., Geometry and convergence of Euler's and Halley's methods, SIAM Rev. 39: 728–735, 1997.
165. Ozban, A.Y, Some new variants of Newton's method, Appl. Math. Lett.17, 677– 682, 2004.
166. Ramanujan, S., Notebooks, 2 Volumes, Tata Institute of Fundamental Research, Bombay, 1957.
167. *Ray, M, Sharma, H.S. and Chaudhary, S, Mathematical Statistics, Ram Prasad and Sons, 2004.*
168. *Ross and Sheldon, M., Introduction to probability and statistics for engineers and scientists*, 4, *Associated Press, 2009.*
169. Sastry, S.S., Introductory Methods of Numerical Analysis. Prentice-Hall, India, 2000.
170. Saxena, H.C., Finite Differences and Numerical Analysis, S. Chand & Company Ltd.

171. Shukla, H.S., Certain Transformation in the Field of Basic Hypergeometric Functions, Ph.D. Thesis, Purvanchal, Jaunpur, 1993.
172. Singh, S.N., Certain transformation of abnormal Basic Hypergeometric Functions, Ramanujan International Symposium on Analysis, Pune, 1987.
173. Singhal, R.P., Transformation formulae for the modified Kampe de Ferieet function, The Mathematics Student, Vol. XLAP, 327-329, 1972.
174. Srivastava, H.M. and Jain, V.K., Some formulas involving q-Jacobi and related polynomials, Annali di Matematica Pura ed Applicata (4) 157, 63–75, 1990.
175. Srivastava, H.M. and Jain, V.K., Some multi linear generating functions for q-Hermite polynomials, Journal of Mathematical Analysis and Applications 144, 147–157, 1989.
176. Srivastava, H.M. and Karlsson, Multiple Gaussian Hypergeometric Series, Ellis Horwood. Ltd. Halsted Press, John Vily & Sons New York, 1985.
177. Srivastava, H.M. and Singhal, J.P., New generating functions for Jacobi and related polynomials, Journal of Mathematical Analysis and Applications 41, 748–752, 1973.
178. Srivastava, H.M., An elementary proof of Bailey's bilinear generating function for Jacobi polynomials, IMA Journal of Applied Mathematics 29, 275–280, 1982.
179. Srivastava, R. and Mathur, K.K., Weighted (0; 0, 2)-interpolation on the Roots of Hermite Polynomials, Acta Mathematica Hungarica 70, 57–73, 1996.
180. Thukral, R., Introduction to a Newton-type method for solving nonlinear equations, Appl. Math. Comp. 195: 663–668, 2008.
181. Traub, J.F., Iterative Methods for the Solution of Equations, Prentice-Hall, Englewood Cliffs, 1964.
182. Verma, A. and Jain, V.K., Some summation formulae of Basic hypergeometric series, Indian J. of Pure and applied Math. 11 (8), 1021-1038, 1980.
183. Verma, A. and Jain, V.K., Some summation formulae of Basic Hypergeometric Series, Indian J. of Pure and applied Math. 11 (8), 1021-1038, 1980.
184. Wang, X. and Liu, L., New eighth-order iterative methods for solving nonlinear equations, J. Comput. Appl. Math. 234: 1611–1620, 2010
185. Watson, G.N., A new proof of Rogers-Ramanujan identities, J. London Math. Soc. 13, 4-9, 1929.
186. Weerakoon, S. and Fernando, A variant of Newton's method with accerated third-order convergence, Appl. Math. Lett, 13: 87–93, 2000.
187. Zarzo, A., Area, I., Godoy, E. and Ronveaux, A., Results for some inversion problems for classical continuous and discrete orthogonal polynomials, Journal of Physics A. Mathematical and General, 30, L35–L40, 1997.
188. Zayed, A.I., Jacobi polynomials as generalized Faber polynomials. Transactions of the American Mathematical Society, 321, 363–378, 1990.
189. Zeng, J., Linearisation de produits de polyn´ omes de Meixner, Krawtchouk, et Charlier.ˆ SIAM Journal on Mathematical Analysis, 21, 1349–1368, 1990.
190. Zeng, J., The q-Stirling numbers, continued fractions and the q-Charlier an Laguerre polynomials, Journal of Computational and Applied Mathematics, 57, 413–424, 1995.
191. Zhang, W., On Chebyshev polynomials and Fibonacci numbers, The Fibonacci Quarterly, 40, 424–428, 2002.
192. Zhang, Z. and Wang, J., On some identities involving the Chebyshev polynomials, The Fibonacci Quarterly, 42, 245–249, 2004.