q-Hypergeometric Series

The q-hypergeometric series are q-analogues of the regular hypergeometric series. The notation for q-hypergeometric series, as for regular hypergeometric series, refers to the number of products in the numerator and denominator over and above $\frac{x^n}{(q;q)_n}$. For example, the q-hypergeometric series with two products in the numerator and one in the denominator is

$${}_2\phi_1(a,b;c;q,x) = \sum_{n=0}^{\infty} \frac{(a;q)_n(b;q)_n}{(c;q)_n} \frac{x^n}{(q;q)_n}.$$

Hypergeometric series are very important in number theory. For example, in the article Sums of Squares we use them in developing polynomials for the sum of two squares and sum of four squares.

The concepts are illustrated using examples from Python programs that use the symbolic programming features of Sympy. For detailed explanations, derivations, and proofs, see Chapter 5 of *An Introduction to q-analysis* by Warren P. Johnson.

1 Hypergeometric Series

These are the q-analogues of hypergeometric series, introduced by Gauss in 1812, which are solutions to differential equations. A hypergeometric series is a power series

$${}_sF_r(a_1,...,a_s;b_1,...,b_r;x) = 1 + \frac{a_1a_2\cdots a_s}{b_1b_2\cdots b_r}\frac{x}{1!} + \frac{a_1(a_1+1)a_2(a_2+1)\cdots a_s(a_s+1)}{b_1(b_1+1)b_2(b_2+1)\cdots b_r(b_r+1)}\frac{x^2}{2!} + ...$$

Defining the rising factorial $(a)_n = a(a+1)(a+2)\cdots a(a+n-1)$, we write

$${}_sF_r(a_1,...,a_s;b_1,...,b_r;x) = \sum_{n=0}^{\infty} \frac{(a_1)_n\cdots(a_s)_n}{(b_1)_n\cdots(b_r)_n}\frac{x^n}{n!}.$$ There are thousands of hypergeometric series.

The simplest examples are ${}_0F_0(;;x) = 1 + \frac{x}{1!} + \frac{x^2}{2!} + ... = e^x$, which is a solution to the differential equation $\frac{dy}{dx} = y$, and ${}_1F_0(a;;x) = \sum_{n=0}^{\infty}(a)_n\frac{x^n}{n!} = 1 + a\frac{x}{1!} + a(a+1)\frac{x^2}{2!} + a(a+1)(a+2)\frac{x^3}{3!}\cdots = (1-x)^{-a}$, which is a solution to the differential equation $\frac{dy}{dx} = x\frac{dy}{dx} + ay$.

2 q-Hypergeometric Series

Analogously, the q-hypergeometric series uses the q-shifted factorial instead of the rising factorial, and is written ${}_{r+1}\phi_r\left(a_1,\ldots,a_{r+1};b_1,\ldots,b_r;q,x\right) = \sum_{n=0}^{\infty}\frac{(a_1;q)_n\cdots(a_{r+1};q)_n}{(b_1;q)_n\cdots(b_r;q)_n}\frac{x^n}{(q;q)_n}$. There are r+1 q-products in the numerator and r products in the denominator, plus one $(q;q)_n$ term.

The simplest q-hypergeometric series is ${}_1\phi_0\left(a;;q,x\right)=\sum_{n=0}^{\infty}\frac{(a;q)_n}{(q;q)_n}x^n$. From the Partition Theory topic, we know that this is the q-binomial series $\sum_{n=0}^{\infty}\frac{(a;q)_n}{(q;q)_n}x^n = \frac{(ax;q)_\infty}{(x;q)_\infty}$.

The next q-hypergeometric series is ${}_2\phi_1\left(a,b;c;q,x\right)$, for which the generic formula is $\sum_{n=0}^{\infty}\frac{(a;q)_n(b;q)_n}{(c;q)_n(q;q)_n}x^n$.

The third q-hypergeometric series is ${}_3\phi_2\left(a,b,c;e,f;q,x\right)$, for which the generic formula is $\sum_{n=0}^{\infty}\frac{(a;q)_n(b;q)_n(c;q)_n}{(e;q)_n(f;q)_n(q;q)_n}x^n$.

3 Heine q-Pfaff Transformation

The general ${}_2\phi_1$ series is transformed with some rewrites and variable substitutions to produce the Heine q-Pfaff transformation:

${}_2\phi_1\left(a,b;c;q,x\right)=\frac{(b;q)_\infty(ax;q)_\infty}{(c;q)_\infty(x;q)_\infty}\,{}_2\phi_1\left(\frac{c}{b},x;ax;q,b\right)$, where all variables are less than 1 and $|c|<|ab|$. To calculate ${}_2\phi_1\left(\frac{c}{b},x;ax;q,b\right)$, we call a Python function for ${}_2\phi_1\left(a,b;c;q,x\right)$ and make a sequence of substitutions. An interesting Python implementation fact is that it is much faster to calculate the symbolic expressions and then substitute numeric values than it is to substitute numeric values and then calculate.

A Python function calculates the two sides of the Heine q-Pfaff transformation over a grid of different values for a, b, c, q, and x. The theorem states that all of the variables must be less than 1. However, the identity has decent accuracy (percent difference of 0.2%) only for very small values of these variables, not just less than 1 but less than 0.005.

4 *q*-Pfaff-Saalschutz Identity

The q-Pfaff-Saalschutz identity uses the ${}_3\phi_2$ function, and one of the several versions is written ${}_3\phi_2\left(q^{-n},a,b;c,\frac{ab}{c}q^{1-n};q,q\right)=\frac{\left(\frac{c}{a};q\right)_n\left(\frac{c}{b};q\right)_n}{(c;q)_n\left(\frac{c}{ab};q\right)_n}$. This is particularly pleasing because it is a balanced q-hypergeometric series. The condition for being balanced is that the product of the numerator variables is the same as the product of the denominator variables, which in this case is $\frac{c}{a}\frac{c}{b}=c\frac{c}{ab}$.

Computationally this identity behaves much better than the Heine q-Pfaff transformation. Across the full range of variable values less than 1, the percent difference is negligible.