Symbolic summation for Mellin space Feynman integrals

M Round
in collaboration with J Blümlein and C Schneider

RISC, Johannes Kepler Universität, Austria.

May, 2017

Computer Algebra

This talk is an overview of some ideas in computer algebra relevant to physics. There are three aims,

1. Discuss the concepts of computer algebra.
2. Outline the literature on summation methods.
3. Details of a modern example of summation relevant to particle physics.

Computer Algebra

Roughly speaking computer algebra is about performing mathematics using algorithms and returning proof certificates. Consider the following example,

$$\sum_{i=0}^{m}(-1)^i\binom{n}{i}$$

An easy way to evaluate the sum is to notice that,

$$\begin{aligned} T(i) &= (-1)^{i+1}\binom{n-1}{i-1} \\ (-1)^i\binom{n}{i} &= T(i+1) - T(i) \\ \Rightarrow \sum_{i=0}^{m}(-1)^i\binom{n}{i} &= \sum_{i=0}^{m} T(i+1) - T(i) \\ &= T(m+1) - T(0) = (-1)^m\binom{n-1}{m} \end{aligned}$$

Computer Algebra

The key statement in our proof is that,

$$(-1)^i \binom{n}{i} = T(i+1) - T(i),$$

which is easy to check using elementary arithmetic. As such, $T(i)$ represents a ‘proof certificate’. $T(i)$ verifies the correctness of the solution independent of how the $T(i)$ was found. Essentially this is what (rigorous) computer algebra is about; using algorithms, implemented on a computer, to obtain expressions with some ‘magical’ property that makes an interesting problem trivial. Our main concern will be with how a $T(i)$ can be found.

Computer Algebra

There is another elementary point worth emphasising. A function, by its nature, contains an infinite amount of information whereas a computer has finite resources. Therefore one must distill the important properties of objects into a finite amount of information; an algebra. If such a setting can be found then there is a hope of obtaining algorithms that can be implemented.

Our setting will be a field $\mathbb{F}$ — a place where one can add, multiply and divide — and hypergeometric sequences taking value in that field,

$$\forall i \in \mathbb{N}\, a_i \in \mathbb{F}, \quad \frac{a_{i+1}}{a_i} = \frac{p(i)}{q(i)}, \quad p(i), q(i) \in \mathbb{F}[i].$$

An example hypergeometric sequence over $\mathbb{Q}[i, n]$ is,

$$a_i = (-1)^i \binom{n}{i}, \quad \frac{a_{i+1}}{a_i} = \frac{i-n}{i+1}.$$

Gosper's Algorithm

Gosper's algorithm is an elementary algorithm that can simplify indefinite sums of a hypergeometric sequence. The basic idea is to use the hypergeometric property to reduce the summation problem to a linear system which is solved using Gaussian elimination. Given a hypergeometric summand t find, whenever possible, a hypergeometric T such that,

$$T(i+1) - T(i) = t(i)$$

Let us just sketch how the algorithm works,

1. Write a_i out as a product of functions with particular properties (always possible, a mere re-writing).
2. By multiplying through by all denominators the problem of finding a hypergeometric T becomes a relation of polynomials. Coefficients comparison leads to a linear system solved by Gaussian elimination. If the system admits no solution then there is no hypergeometric T.

Gosper's Algorithm

Let σ be the shift operator,

$$\sigma(i) = i + 1$$

then Gosper's algorithm amounts to a method for solving the telescoping equation,

$$\sigma(T) - T = t \Rightarrow \sum_{i=0}^{m} t(i) = T(m+1) - T(0).$$

for t the summand of the summation problem. This is our basic strategy for summation problems; solve a telescoping equation so that the sum becomes trivial to solve. (Telescoping was not invented by Gosper, it is a long established technique.) Notice that the telescoping equation is the discrete version of a differential equation,

$$\frac{dY}{dx} = y \Rightarrow \int_a^b y(x)dx = Y(b) - Y(a).$$

Zeilberger's Algorithm

Indefinite summation algorithms are not particularly useful, most difficult problems are definite,

$$S_n = \sum_{i=0}^{n} t(n,i).$$

Summation as a topic in computer algebra began in earnest when Zeilberger gave an algorithm for this problem, again for hypergeometric t. Assume that the result S_n is hypergeometric then there are polynomials,

$$c_0(n)S_n + c_1(n)S_{n+1} = 0$$
$$\Rightarrow \sum_{i=0}^{n} c_0(n)t(n,i) + \sum_{i=0}^{n} c_1(n)t(n+1,i) = -c_1(n)t(n+1,n+1)$$

Zeilberger's idea was to call Gosper's algorithm on,

$$c_0(n)t(n,i) + c_1(n)t(n+1,i).$$

Zeilberger's Algorithm

So far nothing has been achieved; S_n is unknown and so the c_j are too. Zeilberger modified Gosper's algorithm to keep track of internal results and determine for what c_j a hypergeometric solution can exist. Then Zeilberger's algorithm returns specific c_j and a right-hand side f,

$$c_0(n)S_n + c_1(n)S_{n+1}(n) = f(n).$$

It remains to solve for S_n using a recurrence solving technique. I will not discuss that at all except to say M. Petkovsek is a leader in the area and has provided algorithms that can be implemented.

Applications

Automated techniques like Zeilberger's algorithm can be of use in particle physics. It would represent another talk to examine how summation is important for particle physics so let us just make a few observations.

1. Mellin-Barnes integrals are a standard technique for Feynman loop integrals. Applying Cauchy's residue theorem leads to sums over poles, typically involving the Γ-function — which is hypergeometric.
2. Special functions such as Gauß's hypergeometric function need to be expanded in ϵ, the dim. reg. parameter leading to summation problems.
3. Mellin space representations of polylogarithms lead to harmonic sums (my focus).
4. Assorted series expansions and special function identities frequently allow one to convert an integral to a sum.

Applications

Heavy Flavour Wilson Coefficients

The summation problems of interest were found from 3-loop parton distribution functions (PDFs) work,

$$\frac{d^2\sigma}{dxdy} = \frac{2\pi\alpha^2}{xyQ^2}\left[(1+(1-y)^2)F_2(x) - y^2F_L(x)\right],$$

to a product of the (unknown) PDF, f_j, and a perturbative piece; the Wilson coefficients,

$$F_i(x) = \sum_j \int \frac{dz}{z} C_{i,j}\left(\frac{x}{z}\right) f_j(z),$$

$i \in \{2, L\}$ and j runs over all (anti-)quarks and the gluon.
Light flavour contributions are known to 3-loops, heavy contributions to 2-loops.

Recall that the PDF is introduced as a Mellin convolution,

$$F_i(x) = \sum_j \int \frac{dz}{z} C_{i,j}\left(\frac{x}{z}\right) f_j(z) = \sum_j C_{i,j} * f_j.$$

Thus by taking the Mellin transform,

$$\mathcal{M}[g](N) = \int_0^1 dx x^{N-1} g(x) \quad N \in \mathbb{Z}^+$$

things simplify greatly. In Mellin space Feynman diagrams contributing to the Wilson coefficients $C_{i,j}$ become rational in the Mellin parameter N and definite sums involving N. These sums are the primary motivation; to simplify and treat them.

Applications

Illustration (<10 minutes of CPU time)

$$\sum_{i=2}^{n+1}\sum_{j=2}^{n+2-i}\sum_{a=0}^{\infty}\sum_{b=0}^{\infty}\frac{\binom{n+2}{i}\binom{n+2-i}{j}(-1)^{i+j}B[i,j](j+i-1)}{(j+i+a+b)(i+j+a+b-1)(j+b)(i+b)}$$

$$\begin{aligned}
= \quad & \frac{2n^3+5n^2+4n-2}{n+1}S_2+\left[\frac{5}{2}S_2-\frac{n^2+2n+2}{n+1}\right]S_1^2-2nS_{2,1} \\
- \quad & 2n(n+1)\zeta_3-S_1^3+\frac{1}{4}S_1^4-\frac{1}{4}S_2^2+2(n+1)S_3-\frac{1}{2}S_4-2S_{3,1} \\
+ \quad & \left[4\zeta_3+(2n-1)S_2+2S_3-2(n+1)-4S_{2,1}\right]S_1+2S_{2,1,1}
\end{aligned}$$

where $S_{a,\ldots,b}=S_{a,\ldots,b}(n)$ & $B[x,y]=\Gamma[x]\Gamma[y]/\Gamma[x+y]$ for convenience. Notice that the dimensional regularisation parameter is essentially hidden from the talk today. It can be handled with little additional work but it plays no significant role from the computer algebra viewpoint so it's largely hidden today.

Classical Multi-Summation

Consider the problem of simplifying a definite nested m-fold sum using a summation algorithm,

$$S(N) = \sum_{i_1=0}^{N} \sum_{i_2=0}^{i_1} \cdots \sum_{i_m=0}^{i_{m-1}} f(i_1, \dots i_m, N). \tag{1}$$

For concreteness, focus on the scenario where the summand is hypergeometric over a field $\mathbb{F}$ in all arguments,

$$\frac{f(i_1, \dots i_m, N+1)}{f(i_1, \dots i_m, N)} \in \mathbb{F}, \tag{2}$$

and similar for the i_j.

Generically one considers a single sum problem and recursively works outwards. For example,

$$\sum_{i=0}^{N} \sum_{j=0}^{i} \binom{i}{j} = \sum_{i=0}^{N} 2^i = 2^{n+1} - 1. \tag{3}$$

Classical Multi-Summation

$$F_n = \sum_{i_1=\alpha_1}^{L_1(n)} \cdots \sum_{i_m=\alpha_m}^{L_m(n,i_1,\ldots i_m)} \sum_{j_1=\beta_1}^{\infty} \cdots \sum_{j_N=\beta_N}^{\infty} f(n,\{i\},\{j\})$$

Zoom to the innermost sum

Solve Recurrence and substitute in solution

$$\hat{F}_n = \sum_{j_N=\beta_N}^{\infty \text{ or } L_N} f(n,\{i\},\{j\})$$

Find a recurrence for the inner sum,

$$g_0(n)\hat{F}_n + \ldots + g_r(n)\hat{F}_{n+r} = G(n)$$

Refined Multi-Summation

$$F_n = \sum_{i_1=\alpha_1}^{L_1(n)} \cdots \sum_{i_m=\alpha_m}^{L_m(n,i_1,\ldots i_m)} \sum_{j_1=\beta_1}^{\infty} \cdots \sum_{j_N=\beta_N}^{\infty} f(n,\{i\},\{j\})$$

Zoom to the innermost sum

Replace innermost sum with the sequence and its recurrences

$$\hat{F}_n = \sum_{j_N=\beta_N}^{\infty \text{ or } L_N} f(n,\{i\},\{j\})$$

Find a system of recurrences for inner sum, $g_0(n)\hat{F}_n + \ldots + g_r(n)\hat{F}_{n+r} = G(n)$

Telescoping

A standard approach to a single sum with hypergeometric summand is to apply Gosper's or Zeilberger's algorithm depending on the problem. Both algorithms exploit telescoping,

Telescoping (classical & Gosper)

Given a hypergeometric sequence $f(i)$ find a hypergeometric $g(i)$ such that,

$$g(i+1) - g(i) = f(i). \tag{4}$$

Summing over the equation one obtains a simplification for the sum,

$$\sum_{i=1}^{N} f(i) = \sum_{i=1}^{N} g(i+1) - \sum_{i=1}^{N} g(i) = g(N+1) - g(1). \tag{5}$$

The two sums almost exactly cancel each other; the sum telescopes to its first and last points.

Telescoping

A better viewpoint — than thinking of terms cancelling terms — is to think of a discrete version of integration,

$$\int_a^b f(x)dx = g(b) - g(a) \Leftrightarrow g'(x) = f(x), \tag{6}$$

is schematically related to,

$$\sum_{x=a}^{b} f(x) = g(b+1) - g(a) \Leftrightarrow g(x+1) - g(x) = f(x). \tag{7}$$

This will be a useful viewpoint for difference fields later in the talk.

Telescoping

Continuing with the literature: for a definite sum one may use Zeilberger's algorithm which exploits creative telescoping,

Creative Telescoping (Zeilberger)

Given a bivariate hypergeometric sequence $f(N,i)$ find $c_j(N)$ for $j = 0, \ldots, d$ and $g(N,i)$ such that,

$$g(N, i+1) - g(N,i) = c_0(N) f(N,i) + \ldots + c_d(N) f(N+d, i). \tag{8}$$

Summing over the equation one obtains a recurrence for the sum.

$$S_N = \sum_{i=1}^{N} f(N,i),\ \tilde{S}_N(a) = \sum_{i=1}^{a} f(N,i) \Rightarrow \tilde{S}_{N+j}(N) = S_N - \sum_{i=N+1}^{N+j} f(N+j,i)$$

$$g(N, a+1) - g(N,1) = c_0(N)\tilde{S}_N(a) + \ldots + c_d(N)\tilde{S}_{N+d}(a) \tag{9}$$

Colloquially, one finds a linear combination of shifted summands that telescope.

Inhomogeneous Holonomic Systems

Recall the definition of a holonomic sequence,

Holonomic sequence

A sequence a_n is holonomic if there exist polynomials $p_0(x), \ldots, p_r(x) \in \mathbb{F}[x]$ $p_0, p_d \neq 0$ such that,

$$p_0(n)a_n + \ldots + p_d(n)a_{n+d} = 0, \tag{10}$$

for all $n \in \mathbb{N}$.

For refined holonomic summation a generalisation to a quite specific form is needed.

Refined Holonomic System

idea

Build a system from the left-hand sides of a set of holonomic recurrences and allow the right-hand side of the system to be non-zero,

$$\begin{pmatrix} c_0^{(0)} & c_1^{(0)} & \cdots & c_{d-1}^{(0)} \\ c_0^{(1)} & c_1^{(1)} & \cdots & c_{d-1}^{(1)} \\ \vdots & & \vdots & \vdots \\ c_0^{(l)} & c_1^{(l)} & \cdots & c_{d-1}^{(l)} \end{pmatrix} \begin{pmatrix} a_{\vec{n},i} \\ a_{\vec{n},i+1} \\ \vdots \\ a_{\vec{n},i+d-1} \end{pmatrix} + \begin{pmatrix} c_d^{(0)} a_{\vec{n},i+d} \\ c_{\vec{n}+\vec{e}_1}^{(1)} a_{\vec{n}+\vec{e}_1,i} \\ \vdots \\ c_{\vec{n}+\vec{e}_l}^{(l)} a_{\vec{n}+\vec{e}_l,i} \end{pmatrix} = \begin{pmatrix} c_{d+1}^{(0)} \\ c_{d+1}^{(l)} \\ \vdots \\ c_{d+1}^{(l)} \end{pmatrix}$$

for $\vec{e}_l$ unit vectors. This will be referred to as a **refined holonomic system**. Note that the system is of a special but in practice quite general 'hook' form. Secondly, the right-hand side must live where one may do indefinite summation and solve the telescoping problem. This is not quite a definition yet!

Refined Telescoping (first look)

To illustrate what is going on consider the following sum,

$$\sum_{i=0}^{N-10} \sum_{j=0}^{2i} (-1)^j \binom{2i}{j} \binom{N}{i} S_1(j)^2, \quad S_1(j) = \sum_{i=1}^{j} \frac{1}{i}. \tag{11}$$

Follow our approach and compute a recurrence for the inner most sum,

$$\sum_{j=0}^{2i} (-1)^j \binom{2i}{j} S_1(j)^2 \tag{12}$$

The sum obeys a 2-dimensional 'refined holonomic system'. The system is too big to comprehend but let us inspect the form...

Refined Telescoping (first look)

The pure recurrence is,

$$
\begin{aligned}
&-(i+3)(2i+7)^2(32768i^7+446464i^6+2537472i^5+7786752i^4+13909824i^3+14432140i^2+8028567i+1839381)(i+4)^3\mathrm{iSUM}[i+4]\\
&-2(i+3)(1048576i^{12}+31850496i^{11}+437846016i^{10}+3599253504i^9+19687041024i^8+75404168576i^7+207112294816i^6\\
&+410451840512i^5+581501166626i^4+573218302853i^3+372317084504i^2+142659302049i+24299531628)\mathrm{iSUM}[i+3]\\
&-4(3145728i^{13}+97910784i^{12}+1390739456i^{11}+11934441472i^{10}+69013182464i^9+283880729728i^8+854133788960i^7+1901980975328i^6\\
&+3131862368076i^5+3762234911860i^4+3200401706475i^3+1822476616423i^2+621321284001i+95505715980)\mathrm{iSUM}[i+2]\\
&-8(2i+3)^2(1048576i^{11}+27131904i^{10}+313327616i^9+2131001344i^8+9480685568i^7+28958782848i^6+61938013728i^5+92696785984i^4\\
&+95034551942i^3+63466944897i^2+24791126534i+4276165552)\mathrm{iSUM}[i+1]\\
&-64(i+1)^2(2i+1)^2(2i+3)^2\left(32768i^7+675840i^6+5904384i^5+28317952i^4+80507712i^3+135641932i^2+125364847i+49013368\right)\mathrm{iSUM}[i]\\
&=\frac{1}{4}(-81920000i^{11}-2115584000i^{10}-24359116800i^9-164964357120i^8-729590438144i^7-2210988677936i^6-4680628058188i^5\\
&-6914797926836i^4-6977286281717i^3-4572361070022i^2-1747753406411i-294389588076)
\end{aligned}
$$

and the shift in N recurrence,

$$\mathrm{iSUM}[1+N,i]-\mathrm{iSUM}[i]=0.$$

In the refined approach we will try to control the order and objects occurring in this system.

Refined Telescoping (first look)

In the refined approach we generalise the input to be a refined holonomic system and add an f',

Refined telescoping

Given a refined holonomic system for $f(N,i)$ find $c_j(N)$ for $j=0,\ldots,d$, a (possibly zero) $f'(n,i)$ and a $g(N,i)$ such that,

$$g(N,i+1)-g(N,i)+f'(N,i)=c_0(N)f(N,i)+\ldots+c_d(N)f(N+d,i). \quad (13)$$

Holonomic sequences imply we need no closed form for f — f is defined by a recurrence. However a problem may be so hard (more later) that an additional element of freedom f' is added to give flexibility in finding a recurrence. Making sense of f' is the main topic of the talk.

Refined Telescoping

Summing over the equation one obtains an ‘unsimplified’ recurrence for the sum,

$$S_N(a) = \sum_{i=1}^{a} f(N,i), \tag{14}$$

$$g(N,a+1) - g(N,1) = c_0(N)S_N(a) + \ldots + c_d(N)S_{N+d}(a) - \sum_{i=1}^{a} f'(N,i) \tag{15}$$

One may think of the new extension sum,

$$\sum_{i=1}^{a} f'(N,i), \tag{16}$$

as a sub-problem to be solved by the same methods or left unsimplified. This is the ‘price’ one pays for allowing an f'. (f' is NOT a differential!!)

Karr's Difference Theory

To go further we need a proper formalism.

In 1981 Karr gave an indefinite nested sum representation which is an analogue of the Risch algorithm. Karr provided a framework to represent indefinite nested sums and products in difference fields built by a tower of $\Pi\Sigma$-extensions. These ideas were further developed, implemented and fully understood by C Schneider (Schneider '01 '05 see RISC homepage).

Let us briefly review Karr's theory; his main contribution is the ability to allow field extensions like harmonic numbers and this is vital for particle physics.

Karr's Difference Theory

We will not need a detailed understanding of Karr's theory but a flavour of his ideas will be insightful.

Difference field

A difference field $(\mathbb{F}, \sigma)$ is a field $\mathbb{F}$ and a field automorphism $\sigma : \mathbb{F} \to \mathbb{F}$.

The definition is analogous to a differential field. We noted,

$$\int_a^b f(x)dx = g(a) - g(b) \Leftrightarrow \partial g(x) = f. \tag{17}$$

and so with σ corresponding to the derivation of a differential field it is natural that,

$$\sum_{i=a}^{b} f(i) = g(b+1) - g(a) \Leftrightarrow \sigma(g) - g = f. \tag{18}$$

Karr's Difference Theory

The field automorphism, or shift operator, σ encodes how objects behave under $x \mapsto x+1$, a shift in an index. One may denote the index, σ_x, to be clear what index is shifted. To add the harmonic numbers one constructs a Σ-extension s_1 by,

$$S_1(i) = \sum_{j=1}^{i} \frac{1}{j} \Rightarrow \sigma(s_1) = s_1 + \frac{1}{i+1} \Rightarrow s_1 \sim S_1(i) \tag{19}$$

More generally one may add any indefinite nested sum,

$$\sum_{j_1=0}^{i} f_1(j_1) \sum_{j_2=0}^{j_1} f_2(j_2) \cdots \sum_{j_s=0}^{j_{s-1}} f_s(j_s), \tag{20}$$

by constructing a Σ-extension. One may also add Π-extensions,

$$\sigma(p_1) = (i+1)p_1 \Rightarrow p_1 \sim i! = \Gamma(i+1) \tag{21}$$

A $\Pi\Sigma$-field is a tower of such extensions.

Karr's Difference Theory

Conceptually, one would like the shift operator only to affect (sequence) indexed objects and so Karr's construction allows extensions that leave the constants,

$$\mathrm{const}_\sigma(\mathbb{F}) = \{x \in \mathbb{F} | \sigma(x) = x\} \tag{22}$$

unaffected. These are the ΠΣ-extensions, see Schneider '01 '05 for a proper treatment with algorithms etc. (Not important for the talk but a vital concept that underlies algorithms.)
In the difference field one may perform algebraic computations then translate the answer back to the sequence world to express results in terms of sequences.
Today all we need is to recognise one can re-phrase into a ΠΣ-field then (we assume!) algorithms exist.

Refined Holonomic System

Definition

Build a system from the left-hand sides of a set of holonomic recurrences and allow the right-hand side of the system to be non-zero,

$$\begin{pmatrix} c_0^{(0)} & c_1^{(0)} & \cdots & c_{d-1}^{(0)} \\ c_0^{(1)} & c_1^{(1)} & \cdots & c_{d-1}^{(1)} \\ \vdots & & \vdots & \vdots \\ c_0^{(l)} & c_1^{(l)} & \cdots & c_{d-1}^{(l)} \end{pmatrix} \begin{pmatrix} a_{\vec{n},i} \\ a_{\vec{n},i+1} \\ \vdots \\ a_{\vec{n},i+d-1} \end{pmatrix} + \begin{pmatrix} c_d^{(0)} a_{\vec{n},i+d} \\ c_{\vec{n}+\vec{e}_1}^{(1)} a_{\vec{n}+\vec{e}_1,i} \\ \vdots \\ c_{\vec{n}+\vec{e}_l}^{(l)} a_{\vec{n}+\vec{e}_l,i} \end{pmatrix} = \begin{pmatrix} c_{d+1}^{(0)} \\ c_{d+1}^{(l)} \\ \vdots \\ c_{d+1}^{(l)} \end{pmatrix}$$

where $\{c_{d+1}^{(j)}\}$ are elements of a $\Pi\Sigma$-field. (This will be referred to as a refined holonomic system.) For dimensional regularisation there is a field extension $\mathbb{F}[\epsilon]$ which makes little difference however the right-hand side will only be required to a fixed order in ϵ so one can expand to save computer resources.

Karr's Difference Theory

Creative telescoping with recurrences

For example, one can handle the following problem. Define a summand f by a T-**system**: a refined holonomic system of a pure and shifted recurrence and a summand,

$$\begin{aligned} T(N, i+d+1) &= a_0(N)T(N,i) + \ldots + a_d(N)T(N, i+d) + a_{d+1}(N) \\ T(N+1, i) &= b_0(N)T(N, I) + \ldots + b_d(N)T(N, i+d) + b_{d+1}(N) \\ f(N,i) &= h_0(N,i)T(N,i) + \ldots + h_d(N)T(N, i+d) + h_{d+1}(N) \end{aligned}$$

for $h, a, b \in \mathbb{K}(N)(i)$ and possibly non-zero $h_{d+1}, a_{d+1}, b_{d+1}$ in a ΠΣ-field $\mathbb{F} = \mathbb{K}(N)(i)(\{p_a\})(\{s_b\})$. These recurrences may be expressed in a difference field then use algorithms to find a $g \in \mathbb{F}$.

Karr's Difference Theory

Creative telescoping with recurrences

CT with recurrences

Given a T-system for f find a refined holonomic system (summand and shifted recurrences),

$$g(N,i) = g_0(N,i)T(N,i) + \ldots + g_s(N,i)T(N,i+s) + g_{s+1}(N,i) \quad (23)$$

such that,

$$g(N,i+1) - g(N,i) = c_0(N)f(N,i) + \ldots + c_d(N)f(N+d,i). \quad (24)$$

Notice there are no initial conditions on T and so we are not solving a specific problem but the general class of problems that satisfy these recurrences.

Karr's Difference Theory

Creative telescoping with recurrences

If one sums over such a recurrence then one obtains 'telescoping points' which represent a significant overhead in computational problems,

$$\sum_{i=1}^{a}[g(N,i+1)-g(N,i)] = c_0(N)S_N(a)+\ldots+c_d(N)S_{N+d}(a),$$

$$= g_0(N,a)T(N,a)+\ldots+g_s(N,a)T(N,a+s)+g_{s+1}(N,N)$$

$$-\, g_0(N,1)T(N,1)+\ldots+g_s(N,1)T(N,1+s)+g_{s+1}(N,1) \qquad (25)$$

In a multi-sum problem a telescoping point like $T(N,1)$ can be quite an involved summation problem. One would like to control recurrence order to control the number of telescoping points that must be computed.

Inhomogeneous Recurrences

Standard Step

To compute inhomogeneous recurrences (non-zero right-hand side) there are two steps. First apply the classical holonomic approach,

(Specialised) Holonomic Approach

Given a T-system for f compute $g_s(N, i) \in \mathbb{F}$ such that,

$$b_0(N, i)g_s(N, i) + \ldots + b_s(N, i)g_s(N, i+s) = c_1 h_1(N, i) + \ldots + c_d h_d(N, i), \tag{26}$$

where the $\{b_k\}, \{h_k\} \in \mathbb{K}(N)(i)$ are explicitly given (see earlier slides).

Solving this problem gives a solution space where each element contains a g_s and a set of $\{c_k\}$. The lower order g_k are found by back substitution of the g_s into the refined system.

Inhomogeneous Recurrences

RHS Step

If the input recurrences are inhomogeneous then a g_{s+1} is needed,

Step 2

Solve for g_{s+1} and a set $\{\tilde{c}\}$,

$$g_{s+1}(N, i+1) - g_{s+1}(N, i) = \tilde{c}_1 \tilde{h}_1 + \ldots + \tilde{c}_{\tilde{d}} \tilde{h}_{\tilde{d}} \tag{27}$$

where $\{\tilde{h}\}$ are easy to compute and live in $\mathbb{F}$ but depend on the solution chosen from step 1. Similarly $\tilde{d}$ is the dimension of the solution space computed in step 1.

One sees that if one chooses to compute several solutions to step 1 then $\tilde{d}$ grows making it easier to solve the problem. Let us adopt 3 paradigms of solution for g_{s+1}.

Field Extensions

Indefinite Summation

Let us examine Σ-extensions for indefinite summation first. Classical telescoping becomes:

Paradigm 1 (Karr '81)

Let $(\mathbb{F}, \sigma)$ be a $\Pi\Sigma$-field. Given a T-system return a $g_{s+1} \in \mathbb{F}$ such that,

$$\sigma(g_{s+1}) - g_{s+1} = f, \tag{28}$$

or return FAIL if no such g_{s+1} exists.

Using results in difference theory one may determine if a g_{s+1} exists and find it when possible. Algorithms programmed in `Sigma` automate this task (and the next paradigms). If a g_{s+1} exists then the summation problem is solved by summing over the equation as normal. If no such g_{s+1} exists then the problem can not be solved within the construction.

Field Extensions

Indefinite Summation

Suppose no such g_{s+1} exists, then it must be that there exists a field extension, in particular a Σ-extension $(\mathbb{F}[s], \sigma)$ of $(\mathbb{F}, \sigma)$ defined by,

$$\sigma(s) = s + f. \tag{29}$$

This leads to a second option,

Paradigm 2

Let $(\mathbb{F}, \sigma)$ be a $\Pi\Sigma$-field. Given a T-system return a $g_{s+1} \in \mathbb{F}$ such that,

$$\sigma(g_{s+1}) - g_{s+1} = f, \tag{30}$$

or if no such g_{s+1} exists introduce a field extension $\mathbb{F}[s]$ and return $g_{s+1} = s$.

However the solution is crude; one merely enlarges the field to trivially solve the problem. By allowing field extensions low order recurrences can be delivered.

In the refined approach one tries to add more specific, and hopefully intelligent, field extensions by exploiting $\tilde{d} > 1$ solutions.
First construct a big field that contains the solution like in paradigm 2 but with more care. Let $(\mathbb{F}, \sigma)$ be a $\Pi\Sigma$-field then given $f \in \mathbb{F}$ find 'appropriate' Σ-extensions to build an $(\mathbb{F}_k, \sigma)$ such that there exists a g_{s+1} with,

$$\sigma(g_{s+1}) - g_{s+1} + f' = f, \quad f' \in \mathbb{F}_k \tag{31}$$

By 'appropriate' one can be context dependent. This is a strength of the refined approach. One can choose a generic definition such as minimum nested depth or one specific to experience or intuition about a problem. For example one may want to minimise occurrence of a particular sum object etc etc.

Field Extensions

Refined Holonomic Summation

Let $\mathbb{F} = \mathbb{K}(N)(i)(p_1, \ldots p_a)(s_1, \ldots, s_b)$ where the p_i are product extensions and the s_j are sum extensions be the resulting field $\sigma(i) = i + 1$, $\sigma(N) = N$. Then the refined problem becomes

Paradigm 3

Given a T-system for f find a $g_{s+1} \in \mathbb{F}$ and an

$$f' \in \mathbb{F}_k = \mathbb{F}(p_1, \ldots p_a)(s_1, \ldots s_k) \subset \mathbb{F}, \tag{32}$$

with $0 \leq k \leq b$ as small as possible such that,

$$\sigma(g_{s+1}) - g_{s+1} + f' = f. \tag{33}$$

If one can find such a g and f' then summing over (33) gives,

$$g(N+1) - g(0) + \sum_{i=1}^{a} f'(i) = \sum_{i=1}^{a} f(i) \tag{34}$$

Field Extensions

(Definite) Refined Holonomic Summation

Thus far only field extensions for indefinite summation have been discussed. Going to definite summation introduces a sequence of $c_j(N)$ as before. This leads to,

Definite Refined Summation

Given a T-system find a $g \in \mathbb{F}$, $c_j(N) \in \mathbb{K}(N)$ for $j = 0, \ldots, d$ and an,

$$f' \in \mathbb{F}_k = \mathbb{F}(p_1, \ldots, p_a)(s_1, \ldots, s_k) \subset \mathbb{F}, \tag{35}$$

such that,

$$\sigma(g) - g + f' = \sum_{m=1}^{\Delta} c_m(N) f_{N+m}, \tag{36}$$

with k and $\tilde{d}$ as small as possible and f_{N+m} denoting a sequence of objects created from shifts in N of f into the difference field.

Trading extensions

Suppose one picks a recurrence order d and there exists a g and c_j. That fixes a vector space constructed over the $c_j(N)$.

1. As one increases d in step 1 the vector space of solutions gets bigger and the chances of finding an $f' \in \mathbb{F}_k$ increase, giving a refined solution.
2. Assuming the sum is a holonomic sequence then there will exist a d such that $f' = 0$. (paradigm 1).
3. If paradigm 1 does not hold then paradigm 2 must which defines $f' = s$, where s was the maximal extension from before.

Thus one must find an f' and it may 'sit between' paradigms 1 & 2. In this sense one can trade extensions for recurrence order.

Refined Multi-Summation

$$F_n = \sum_{i_1=\alpha_1}^{L_1(n)} \cdots \sum_{i_m=\alpha_m}^{L_m(n,i_1,\ldots i_m)} \sum_{j_1=\beta_1}^{\infty} \cdots \sum_{j_N=\beta_N}^{\infty} f(n,\{i\},\{j\})$$

Zoom to the innermost sum

Replace innermost sum with the sequence and its recurrences

$$\hat{F}_n = \sum_{j_N=\beta_N}^{\infty \text{ or } L_N} f(n,\{i\},\{j\})$$

Find a system of recurrences for inner sum, $g_0(n)\hat{F}_n + \ldots + g_r(n)\hat{F}_{n+r} \;=\; G(n)$

Multi-sums

Now we can return to multi-sums using refined holonomic summation.

1. Given a multi-sum one may go to the inner most sum and compute a recurrence using the the freedom of the refined approach to spread the weight of the problem across both recurrence order and the number of field extensions.
2. Without solving the resulting recurrence, one may directly proceed to the next sum.
3. After working outwards one will deliver a recurrence for the entire sum. The computation time and memory requirements can be controlled by trading recurrences and field extensions.

Worked Examples

A lot of ideas and theory has been covered! To illustrate the features it is worth looking at some example multi-sums, summands defined by recurrences and recurrences with 'unsimplified' sums in them.

Worked Examples

Example 1 (Paradigm 1)

Consider the following definite nested multi-sum,

$$S_N = \sum_{i=0}^{N-3} \sum_{j=0}^{N-i-2} \frac{4i!(-1)^j (N-i-1)(1+j)!}{(1+i+j)^2(2+i+j)^2(i+j)!} \binom{N-i-2}{j}$$

To find a recurrence for S_N, first re-write the sum,

$$4 \sum_{i_1=0}^{N-2} i!(N-i-1) a_{i,N}$$

$$a_{i,N} = \sum_{j=0}^{N-i-2} \frac{(-1)^j (1+j)!}{(1+i+j)^2(2+i+j)^2(i+j)!} \binom{N-i-2}{j}$$

now apply creative telescoping over holonomic sequences e.g. with `Sigma`.

Worked Examples

Example 1 (Paradigm 1)

In[1]:= $<<$ "Sigma.m";

In[2]:= $\mathbf{X_1 = \frac{(-1)^j(1+j)!}{(1+i+j)^2(2+i+j)^2(i+j)!} Binomial[N-i-2, j];}$

In[3]:= $\mathbf{sum1 = SigmaSum[X_1, \{j, 0, N-i-2\}];}$

In[4]:= $\mathbf{PRec = First[GenerateRecurrence[sum1, i]]/.SUM- > iSUM}$

Out[4]= $(2+i+N+iN)iSUM[i]-(2+i-N)(1+i+N)iSUM[i+1] = \frac{1}{(1+i^2)i!}$

In[5]:= $\mathbf{PRec = \{PRec, iSUM[i]\};}$

In[6]:= $\mathbf{SRec = First[GenerateRecurrence[sum1, i, OneShiftIn- > N]]/.SUM- > iSUM}$

Out[6]= $-(1+i-N)(1+2i+iN)iSUM[i]-(1+N)(1+i+i_1N)iSUM[N+1, i] = -\frac{1}{(1+N)i!}$

In[7]:= $\mathbf{GenerateRecurrence[\sum_{i=0}^{N-2} 4i!(N-i-1)iSUM[i], N, PRec, SRec]}$

Worked Examples

Example 1 (Paradigm 1)

`Sigma` returns an unsimplified recurrence,

$$\begin{aligned}\mathrm{SUM}(N) = &-\frac{4\sum_{i=1}^{N-3}\frac{1}{i}}{(N+1)(N+2)} + \frac{4\mathrm{iSUM}[0]\left(N^2-N\right)}{(N+1)(N+2)} \\ &+ \frac{8(N-2)\left(N^3-3N-1\right)\mathrm{iSUM}[N-3](N-3)!}{(N+1)(N+2)\left(N^2-2N-2\right)} \\ &+ \frac{4\left(N^4-2N^3-2N^2-3N\right)}{(N+1)(N+2)\left(N^2-2N-2\right)}\end{aligned} \tag{37}$$

The telescoping points and may be calculated from the definition

$$\mathrm{iSUM}[i] = \sum_{j=0}^{N-i-2}\frac{(-1)^j(1+j)!}{(1+i+j)^2(2+i+j)^2(i+j)!}\binom{N-i-2}{j} \tag{38}$$

as separate summation problems. In addition one can recognise the harmonic numbers in the result.

Combining everything and some tidying up one obtains a final answer,

$$S_N = \frac{N^5 - 2N^4 + N^3 - N - 2}{(N-1)^2 N^2 (2+N)} - \frac{2S_1(N)}{(1+N)(2+N)}.$$

More generally, one would like to balance the number of extensions sums (next example) to calculate and the number of telescoping points to calculate. In practical implementations the time to simplify a right-hand side by computing these sums is typically greater than finding the recurrence.

Finally an example where the refined approach can be neither paradigm 1 or 2. Consider the following sum,

$$\sum_{i=0}^{N-10} \sum_{j=0}^{2i} (-1)^j \binom{2i}{j} \binom{N}{i} S_1(j)^2 \tag{39}$$

Here holonomic summation techniques (paradigm 1) succeed for the inner sum,

```
In[8]:= sum3 = SigmaSum[(−1)^j Binomial[2i, j]Binomial[N, i]S[1, j]^2, j, 0, 2i]
In[9]:= Prec3 = GenerateRecurrence[sum3, i, SimplifyByExt− > None];
```

(Using S for the Harmonic numbers is traditional in particle physics.)

Worked Examples

Example 2 (All Paradigms)

The returned recurrence is inhomogeneous and with no field extensions. Paradigm 2 is not needed because paradigm 1 succeeds. However the resulting recurrence is 4^{th} order and too big to show. One can find something more compact by asking `Sigma` to use refined theory,

In[10]:= **Prec3DN = GenerateRecurrence[sum3, i, SimplifyByExt− > DepthNumber];**

`Sigma` computes a 3^{rd} order recurrence with two field-extensions,

$$s_1 = \sum_{j=0}^{2i} \frac{(-1)^j \binom{2i}{j}^2}{1 + 2i - j}, \qquad s_2 = \sum_{j=0}^{2i} \frac{(-1)^j \binom{2i}{j}^2}{2 + 2i - j}. \tag{40}$$

The ‘price’ of using refined difference theory is that both extension sums represent separate summation problems. One can find with `Sigma` for example,

$$s_1 = \frac{(-1)^i(2i)!}{(1+2i)(i!)^2}, \qquad s_2 = \frac{(-1)^i(2i)!(1+4i+2i^2)}{2(1+i)^2(1+2i)(i!)^2}. \tag{41}$$

One can substitute back into the recurrence and obtain something still quite large,

Worked Examples

Example 2 (All Paradigms)

$$
\begin{aligned}
&2(i+1)\left(32i^3+116i^2+130i+43\right)\mathrm{iSUM}(i+1)(i-N+1)\\
&-4(2i+1)^2(8i+13)\mathrm{iSUM}(i)(i-N)(i-N+1)\\
&-(i+1)(8i+5)(i+2)^3\mathrm{iSUM}(i+2)\\
&=r(i,N)\binom{N}{i} \qquad (42)
\end{aligned}
$$

for $r(i,N)$ a large rational expression in i and N. Now we go to the outer sum which requires a shifted recurrence,

In[11]:= $\mathbf{Srec3 = GenerateRecurrence[sum3, i, OneShiftIn- > N]}$

Out[11]= $(1+N)iSUM[i]+(-N+i-1)iSUM[1+N,i]==0$

Worked Examples

Example 2 (All Paradigms)

There are now quite a few options to compute the final recurrence.

1. Using the 4^{th} order inner sum recurrence one can obtain a 10^{th} order recurrence with no extensions (paradigm 1) in 110s.
2. Or, using paradigm 2 via SimplifyByExt$-$ $>$Full, a 4^{th} order recurrence with 8 extension sums in 64s.
3. Using the 3^{rd} order recurrence paradigm 1 now finds an 8^{th} order recurrence in 38s.
4. Finally, using paradigm 2 via SimplifyByExt$-$ $>$Full there is a 4^{th} order recurrence with 5 extension sums that an be found in 15s.

Using holonomic recurrences throughout generally leads to large computation times (large linear systems) with lots of telescoping points while the refined approach is usually more efficient for recurrences but with the additional processing of extension sums.

Worked Examples

Example 2 (All Paradigms)

The 5 extension sums found in the last configuration are,

$$\sum_{i=0}^{N-10} \frac{\binom{N}{i}}{1+N-i}, \qquad \sum_{i=0}^{N-10} \frac{\binom{N}{i}}{1+2i},$$
$$\sum_{i=0}^{N-10} \frac{(-1)^i (2i)!}{(1+N-i)(i!)^2} \binom{N}{i}, \quad \sum_{i=0}^{N-10} \frac{(-1)^i i (2i)!}{(1+N-i)(2i-1)(i!)^2} \binom{N}{i},$$
$$\sum_{i=0}^{N-10} \frac{(-1)^i (2i)!}{(2i+1)(i!)^2} \binom{N}{i}. \tag{43}$$

All of these sums are hypergeometric and so a parallel application of Sigma or Zeilberger's algorithm etc can simplify these sums.

ρSum Implementation

An implementation of the refined approach using `Sigma` to find recurrences has been produced known as ρSum. It involves choices of paradigm type with various limits on time and recurrence order. The implementation was tuned towards a specific problem in particle physics that led to a specific class of sums. Intuition and experience of the problem guided the way in which the code choosing between the various types of recurrence to compute.

Summary

1. The features of refined holonomic summation have been outlined as a concept in difference field theory.
2. In particular on may trade field extensions and recurrence order.
3. For multi-summation problems one can exploit the freedom of recurrences in problems where holonomic approaches are cumbersome.
4. Refined summation can be significantly faster and use less memory than standard holonomic approaches.