Guessing Handbook

Manuel Kauers, RISC–Linz.

Version 0.32, 2009–04–07 / supported by FWF grants P19462–N18 and P20162–N18.

```
In[1]:= << Guess.m
```

```
Guess  Package by Manuel Kauers – © RISC Linz – V 0.32 2009-04-07
```

Introduction

This package provides functions for discoverying linear recurrence equations with polynomial coefficients satisfied by the entries of a given array. It extends the functionality of existing packages like Mallinger's **GeneratingFunctions.m** or Salvy and Zimmermann's **gfun** in two directions: (1) it is *more general* in that, e.g., multivariate recurrences, or recurrences of a particular shape can be searched for. (2) it is *more efficient* than the corresponding functions of existing packages, by using fine tuned linear system solvers based on modular techniques.

The basic principle of all functions provided in this package is the same: they take an array of numbers as input and return as output a set of recurrence equations these numbers satisfy. For example, one way to rediscover the Fibonacci recurrence is to type

```
In[2]:= GuessMinRE[{1, 1, 2, 3, 5, 8, 13, 21, 34}, f[n]]
```

```
Out[2]= -f[n] - f[1 + n] + f[2 + n]
```

This is a recurrence with constant coefficients, and it can also be found by Mathematica's builtin function **FindLinearRecurrence** (available since Mathematica 7). The builtin function, however, is not able to discover recurrence equations with polynomial coefficients, like the recurrence for the harmonic numbers, which our package can find without any problem:

```
In[3]:= GuessMinRE[{0, 1, 3/2, 11/6, 25/12, 137/60, 49/20, 363/140, 761/280, 7129/2520, 7381/2520}, f[n]]
```

```
Out[3]= (1 + n) f[n] + (-3 - 2 n) f[1 + n] + (2 + n) f[2 + n]
```

Every finite array of numbers satisfies such recurrence, in fact, an infinite dimensional vector space of recurrence equations. But most of these are not interesting. The package was written with applications from combinatorics and experimental mathematics in mind. In these applications, the finite arrays provided as input belong to an infinite sequence of numbers (like the sequence of Fibonacci numbers or harmonic numbers above), and one is interested only in recurrence equations which are valid for the entire sequence, not just for the given prefix.

It is clear that a program has no chance to know how the a sequence continues beyond the given data. But it is possible to distinguish in the set of all recurrence equations satisfied by a finite list of numbers among those which hold "for generic reasons" and those which are somehow special. Those which are special are, with high probability, valid for the entire sequence, while the generic ones are just artefacts which are caused by the truncration of the sequence to a finite array. The functions of this package are made such that they ignore the "generic" recurrences as much as possible and only present "special" recurrences as output. It is therefore fair to regard the output of our functions as *guessed* recurrences for the entire sequences whose prefixes had been given as input.

In this document, we describe the usage of the functions provided. There is one, rather elaborate, function available for guessing multivariate recurrence equations of multivariate sequences. This function is described in the next section. In the section afterwards, we describe a collection of special purpose functions custom–tailored for guessing univariate recurrence equations or differential equations or algebraic equations.

GuessMultRE: Multivariate Guessing

■ Overview

The function takes an array of numbers (possibly multidimensional) as input, as well as a description of the terms which may occur in the recurrence to be searched. It then outputs all the recurrence equations satisfied by the given data which are linear combinations of the specified sets of terms. Terms are of the form $n^{\alpha} f[n + \beta]$, where n, α and β are understood as multi–indices. But terms are not specified in this way, this would be too cumbersome. Instead, only the parts of the form $f[n + \beta]$ have to be specified explicitly as a list (this is called the structure set). The terms n^{α} in the polynomial prefactor are given just by the list of variables and a bound on the total degree.

Example:

```
In[4]:= data = Table[n ! 2^k, {n, 0, 5}, {k, 0, 5}]

Out[4]= {{1, 2, 4, 8, 16, 32}, {1, 2, 4, 8, 16, 32}, {2, 4, 8, 16, 32, 64},
         {6, 12, 24, 48, 96, 192}, {24, 48, 96, 192, 384, 768}, {120, 240, 480, 960, 1920, 3840}}

In[5]:= GuessMultRE[data, {f[n, k], f[n + 1, k], f[n, k + 1]}, {n, k}, 1]

Out[5]= {-(1 + n) f[n, k] + f[1 + n, k], -2 n f[n, k] + n f[n, 1 + k],
         -2 k f[n, k] + k f[n, 1 + k], -2 f[n, k] + f[n, 1 + k]}
```

This gives *all* the recurrences of the given array which can be expressed as linear combinations of the following terms: $\{f[n, k], k f[n, k], n f[n, k], f[n, 1 + k], k f[n, 1 + k], n f[n, 1 + k], f[1 + n, k], k f[1 + n, k], n f[1 + n, k]\}$. The actual set of terms can be accustumed in various ways. See below for details.

Note: There may be additional recurrences satisfied by the given data, but if they are not linear combinations of the ones output by the function, then they must involve additional terms. This way, the existence of recurrences of a certain form can be explicitly excluded. For example, the computation

```
In[6]:= GuessMultRE[{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31}, {f[n], f[n + 1], f[n + 2]}, {n}, 2]

Out[6]= {}
```

constitutes a *rigorous proof* that the prime numbers do not satisfy a second order linear recurrence equation with quadratic coefficients. On the other hand, if a recurrence had been discovered, this does *not* guarantee that this recurrence is valid for all the primes, it it just guaranteed that the recurrence is valid for the finite array of numbers given in the input.

Note also: Big recurrences can be only found if there is enough data supplied. Roughly, there must be more elements in the array than the number of terms allowed in the recurrence. In the previous example with the primes, we have given 11 primes and allowed 9 terms for the recurrence. If we seek a recurrence with polynomial coefficients of degree 3, then the first 11 primes won't be sufficient for the system to distinguish good recurrences from bad ones, it will therefore complain:

```
In[7]:= GuessMultRE[{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31}, {f[n], f[n + 1], f[n + 2]}, {n}, 3]

    Throw::nocatch : Uncaught Throw[insufficient input data.] returned to top level. >>

Out[7]= Hold[Throw[insufficient input data.]]
```

■ Parameters

■ Data Array

The data array containing the prefix of the sequence must be rectangular (triangular arrays are not allowed), but may have any dimension. The terms specified in the structure set as well as the number of variables must match the dimension of the array. The elements of the array may be rational numbers, but may as well belong to an algebraic number field (see comments on option **Extension**), or have parameters, or belong to a finite field (see comments on option **Modulus**).

Example 1D: (Note that for 1D arrays, the package provides more efficient special purpose functions, which are described separately.)

```
In[8]:= GuessMultRE[{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, {f[n], f[n + 1]}, {n}, 1]

Out[8]= {-(2 + n) f[n] + (1 + n) f[1 + n]}
```

Example 2D:

```
In[9]:= GuessMultRE[{{1, 2, 4, 8, 16, 32}, {1, 2, 4, 8, 16, 32},
          {2, 4, 8, 16, 32, 64}, {6, 12, 24, 48, 96, 192}, {24, 48, 96, 192, 384, 768},
          {120, 240, 480, 960, 1920, 3840}}, {f[n, k], f[n + 1, k]}, {n, k}, 1]

Out[9]= {-(1 + n) f[n, k] + f[1 + n, k]}
```

Example 3D:

```
In[10]:= data = Table[JacobiP[n, a, b, 3], {n, 0, 5}, {a, 0, 5}, {b, 0, 5}]

Out[10]= {{{1, 1, 1, 1, 1, 1}, {1, 1, 1, 1, 1, 1}, {1, 1, 1, 1, 1, 1}, {1, 1, 1, 1, 1, 1}, {1, 1, 1, 1, 1, 1},
           {1, 1, 1, 1, 1, 1}}, {{3, 4, 5, 6, 7, 8}, {5, 6, 7, 8, 9, 10}, {7, 8, 9, 10, 11, 12},
           {9, 10, 11, 12, 13, 14}, {11, 12, 13, 14, 15, 16}, {13, 14, 15, 16, 17, 18}},
          {{13, 19, 26, 34, 43, 53}, {25, 33, 42, 52, 63, 75}, {41, 51, 62, 74, 87, 101},
           {61, 73, 86, 100, 115, 131}, {85, 99, 114, 130, 147, 165}, {113, 129, 146, 164, 183, 203}},
          {{63, 96, 138, 190, 253, 328}, {129, 180, 242, 316, 403, 504},
           {231, 304, 390, 490, 605, 736}, {377, 476, 590, 720, 867, 1032},
           {575, 704, 850, 1014, 1197, 1400}, {833, 996, 1178, 1380, 1603, 1848}},
          {{321, 501, 743, 1059, 1462, 1966}, {681, 985, 1375, 1865, 2470, 3206},
           {1289, 1765, 2355, 3075, 3942, 4974}, {2241, 2945, 3795, 4809, 6006, 7406},
           {3649, 4645, 5823, 7203, 8806, 10 654}, {5641, 7001, 8583, 10 409, 12 502, 14 886}},
          {{1683, 2668, 4043, 5908, 8378, 11 584}, {3653, 5418, 7773, 10 848, 14 790, 19 764},
           {7183, 10 128, 13 923, 18 732, 24 738, 32 144}, {13 073, 17 718, 23 541, 30 744, 39 550, 50 204},
           {22 363, 29 364, 37 947, 48 356, 60 858, 75 744}, {36 365, 46 530, 58 765, 73 360, 90 630, 110 916}}}

In[11]:= GuessMultRE[data, {f[n, a, b], f[n + 1, a, b], f[n + 2, a, b], f[n, a + 1, b], f[n + 1, a + 1, b],
           f[n + 2, a + 1, b], f[n, a, b + 1], f[n + 1, a, b + 1], f[n + 2, a, b + 1]}, {n, a, b}, 0]

Out[11]= {f[2 + n, a, b] - 2 f[2 + n, a, 1 + b] + f[2 + n, 1 + a, b],
          f[1 + n, a, b] - 2 f[1 + n, a, 1 + b] + f[1 + n, 1 + a, b], f[n, a, b] - 2 f[n, a, 1 + b] + f[n, 1 + a, b]}
```

■ Structure Set

The structure set declares the shifts which may occur in the desired recurrence equation. Shift operators are represented using a function symbol, e.g., $f[n + 1]$ represents the shift in n. The choice of the function symbol is up to the user. The arity of the funciton symbol must be the same in all elements of the structure set, and the ith argument must be of the form $n_i + k_i$ where n_i is the ith variable (see the section on variable declaration below) and k_i is a nonnegative integer. We have deliberately enforced some reduncance in the input specification, in order to be able to detect faulty input by inconsistenies, so that lengthy computations as a consequence of typos become less likely.

The structure set may also contain the element 1, in order to allow for the detection of inhomogeneous recurrence equations. The recurrence equations to be determined are all of the form $p_1 f_1 + p_2 f_2 + p_3 f_3 + \ldots$, where the p_i are polynomials whose possible terms are specified elsewhere, and the f_i are the elements of the structure set specified here.

Example:

```
In[12]:= data = Table[JacobiP[n, a, b, 3], {n, 0, 5}, {a, 0, 5}, {b, 0, 5}];
```

Search for recurrence equation with shifts in n only:

```
In[13]:= GuessMultRE[data, {f[n, a, b], f[n + 1, a, b], f[n + 2, a, b]}, {n, a, b}, 3]
```

$$\text{Out[13]=}\ \Big\{\frac{1}{2}(1+a+n)(1+b+n)(4+a+b+2n)\,f[n,a,b]-\frac{1}{2}(3+a+b+2n)\left(12+9a+2a^2+9b+3ab+b^2+18n+6an+6bn+6n^2\right)f[1+n,a,b]+\frac{1}{2}(2+n)(2+a+b+n)(2+a+b+2n)\,f[2+n,a,b]\Big\}$$

Search instead for a recurrence equation with shifts in a only, or with shifts in b only:

```
In[14]:= GuessMultRE[data, {f[n, a, b], f[n, a + 1, b], f[n, a + 2, b]}, {n, a, b}, 1]
```

$$\text{Out[14]=}\ \{-(1+a+n)\,f[n,a,b]-(1+b+2n)\,f[n,1+a,b]+(2+a+b+n)\,f[n,2+a,b]\}$$

```
In[15]:= GuessMultRE[data, {f[n, a, b], f[n, a, b + 1], f[n, a, b + 2]}, {n, a, b}, 1]
```

$$\text{Out[15]=}\ \Big\{\frac{1}{2}(1+b+n)\,f[n,a,b]-\frac{1}{2}(5+2a+3b+4n)\,f[n,a,1+b]+(2+a+b+n)\,f[n,a,2+b]\Big\}$$

Or search for a mixed recurrence, with shifts in n, a, b :

```
In[16]:= sset = Flatten[Table[f[n + u, a + v, b + w], {u, 0, 1}, {v, 0, 1}, {w, 0, 1}]]
```

```
Out[16]= {f[n, a, b], f[n, a, 1 + b], f[n, 1 + a, b], f[n, 1 + a, 1 + b],
  f[1 + n, a, b], f[1 + n, a, 1 + b], f[1 + n, 1 + a, b], f[1 + n, 1 + a, 1 + b]}
```

```
In[17]:= GuessMultRE[data, sset, {n, a, b}, 1]
```

```
Out[17]= {5 (1 + b + n) f[n, a, b] - (17 + 7 a + 10 b + 17 n) f[n, a, 1 + b] +
    (2 a + b) f[n, 1 + a, 1 + b] - (2 + a + b) f[1 + n, a, b] + (3 + a + b + n) f[1 + n, 1 + a, 1 + b],
   3 (1 + b + n) f[n, a, b] - 2 (5 + 2 a + 3 b + 5 n) f[n, a, 1 + b] + 2 (2 + a + b) f[n, 1 + a, 1 + b] +
    f[1 + n, a, b] + n f[1 + n, 1 + a, b], -2 b f[n, 1 + a, 1 + b] - b f[1 + n, a, b] + b f[1 + n, 1 + a, b],
   -2 a f[n, 1 + a, 1 + b] - a f[1 + n, a, b] + a f[1 + n, 1 + a, b],
   -2 f[n, 1 + a, 1 + b] - f[1 + n, a, b] + f[1 + n, 1 + a, b],
   2 (1 + b + n) f[n, a, b] - (7 + 3 a + 4 b + 7 n) f[n, a, 1 + b] + (2 + a + b) f[n, 1 + a, 1 + b] +
    f[1 + n, a, b] + n f[1 + n, a, 1 + b], -b f[n, 1 + a, 1 + b] - b f[1 + n, a, b] + b f[1 + n, a, 1 + b],
   -a f[n, 1 + a, 1 + b] - a f[1 + n, a, b] + a f[1 + n, a, 1 + b],
   -f[n, 1 + a, 1 + b] - f[1 + n, a, b] + f[1 + n, a, 1 + b],
   (1 + b + n) f[n, a, b] - 2 (2 + a + b + 2 n) f[n, a, 1 + b] + (1 + n) f[1 + n, a, b],
   (1 + b + n) f[n, a, b] - (3 + a + 2 b + 3 n) f[n, a, 1 + b] + (2 + a + b + n) f[n, 1 + a, 1 + b],
   n f[n, a, b] - 2 n f[n, a, 1 + b] + n f[n, 1 + a, b], b f[n, a, b] - 2 b f[n, a, 1 + b] + b f[n, 1 + a, b],
   a f[n, a, b] - 2 a f[n, a, 1 + b] + a f[n, 1 + a, b], f[n, a, b] - 2 f[n, a, 1 + b] + f[n, 1 + a, b]}
```

Or search for a mixed recurrence, with shifts in a, b, but not in n, and shifts in a and b subjected to the restriction that they must have the same parity, and not exceed 2 in total degree:

```
In[18]:= sset = DeleteCases[Flatten[
           Table[If[EvenQ[v + w], f[n + 2 u, a + v, b + w]], {u, 0, 0}, {v, 0, 2}, {w, 0, 2 - v}]], Null]
```

```
Out[18]= {f[n, a, b], f[n, a, 2 + b], f[n, 1 + a, 1 + b], f[n, 2 + a, b]}
```

```
In[19]:= GuessMultRE[data, sset, {n, a, b}, 1]
```

$$\text{Out[19]=}\ \Big\{(1+b)\,f[n,a,b]-2(3+a+2b+n)\,f[n,a,2+b]+(5+2a+3b)\,f[n,1+a,1+b]+n\,f[n,2+a,b],\ -b\,f[n,a,b]+4b\,f[n,a,2+b]-4b\,f[n,1+a,1+b]+b\,f[n,2+a,b],\ -a\,f[n,a,b]+4a\,f[n,a,2+b]-4a\,f[n,1+a,1+b]+a\,f[n,2+a,b],\ -f[n,a,b]+4f[n,a,2+b]-4f[n,1+a,1+b]+f[n,2+a,b],\ \frac{1}{4}(1+b+n)\,f[n,a,b]-\frac{1}{2}(3+a+2b+3n)\,f[n,a,2+b]+\frac{1}{4}(5+2a+3b+4n)\,f[n,1+a,1+b]\Big\}$$

■ Variables

As a third argument, the **GuessMultRE** command expects the list of variables to be used for the polynomial coefficients (and also as arguments for the terms in the structure set, see above). The number of variables must be equal to the number of dimensions of the **data** array supplied as first argument, and the *i*th argument corresponds to the *i*th dimension of the array, so that $f[n, m, k, ...]$ corresponds to **data[[n+1,m+1,k+1,...]]** etc. (see Option **StartPoint** for variations).

It is possible to declare *blocks* of variables. This is useful if there are symmetries in the given data, e.g., if it is known that $f[n, k] = f[k, n]$ for the sequence under consideration. In this case, it can often be observed that whenever there exists a recurrence for a certain structure set with polynomial coefficients of a certain degree, then there also exists a recurrence for the same structure set with *symmetric polynomials* of the same degree. Therefore, it suffices to search for such recurrence equations, and ignore all others. This speeds up the computation and also reduces the size of the output.

Variable blocks are specified by additional braces **{}** around the variables to be blocked together. For instance, **{{n,k},{m}}** would declare a symmetry between n and k, while m stands for itself. The default variable declaration without blocks, e.g., **{n,k,m}** is equivalent to a variable declaration where every variable is a block of itself: **{{n},{k},{m}}**.

Example.

```
In[20]:= data = CoefficientList[
           Normal[Series[1 / (1 - x - y - z + 4 x y z), {x, 0, 15}, {y, 0, 15}, {z, 0, 15}]], {x, y, z}];

In[21]:= Timing[GuessMultRE[data, {f[n, k, m], f[n + 1, k + 1, m + 1], f[n + 2, k + 2, m + 2]}, {n, k, m}, 4]]
```

$$\text{Out[21]=}\ \Big\{0.776048,\ \Big\{8\,(-1+k-m-n)\,(1+k+m-n)\,(1+k-m+n)\,(6+k+m+n)\,f[n,k,m]-\big(96+90\,k+23\,k^2+6\,k^3+k^4+90\,m+70\,k\,m+6\,k^2\,m+23\,m^2+6\,k\,m^2-2\,k^2\,m^2+6\,m^3+m^4+90\,n+70\,k\,n+6\,k^2\,n+70\,m\,n+72\,k\,m\,n+8\,k^2\,m\,n+6\,m^2\,n+8\,k\,m^2\,n+23\,n^2+6\,k\,n^2-2\,k^2\,n^2+6\,m\,n^2+8\,k\,m\,n^2-2\,m^2\,n^2+6\,n^3+n^4\big)\,f[1+n,1+k,1+m]+(2+k)\,(2+m)\,(2+n)\,(3+k+m+n)\,f[2+n,2+k,2+m]\Big\}\Big\}$$

```
In[22]:= Timing[GuessMultRE[data, {f[n, k, m], f[n + 1, k + 1, m + 1], f[n + 2, k + 2, m + 2]}, {{n, k, m}}, 4]]
```

$$\text{Out[22]=}\ \Big\{0.204013,\ \Big\{4\,(-1+k-m-n)\,(1+k+m-n)\,(1+k-m+n)\,(6+k+m+n)\,f[n,k,m]-\frac{1}{2}\big(96+90\,k+23\,k^2+6\,k^3+k^4+90\,m+70\,k\,m+6\,k^2\,m+23\,m^2+6\,k\,m^2-2\,k^2\,m^2+6\,m^3+m^4+90\,n+70\,k\,n+6\,k^2\,n+70\,m\,n+72\,k\,m\,n+8\,k^2\,m\,n+6\,m^2\,n+8\,k\,m^2\,n+23\,n^2+6\,k\,n^2-2\,k^2\,n^2+6\,m\,n^2+8\,k\,m\,n^2-2\,m^2\,n^2+6\,n^3+n^4\big)\,f[1+n,1+k,1+m]+\frac{1}{2}(2+k)\,(2+m)\,(2+n)\,(3+k+m+n)\,f[2+n,2+k,2+m]\Big\}\Big\}$$

If a variable is given in the form q^n, then the q−shift is used instead of the ordinary shift. Here, q may either be a symbol (like **q**) or a number different from 1 (like 2). It is also allowed to use different bases for different variables, or to mix ordinary shifts with q−shifts.

Example.

```
In[23]:= GuessMultRE[Table[Product[(1 + q^i) / (1 + 2 q^(i + 1)), {i, 1, n}], {n, 1, 20}],
          {f[n], f[n + 1]}, {q^n}, 1]
```

$$\text{Out[23]=}\ \Big\{-\frac{1}{2}\left(1+q^{2+n}\right)f[n]+\frac{1}{2}\left(1+2\,q^{3+n}\right)f[1+n]\Big\}$$

■ Degree

The degree is usually just an integer which bounds the total degree of the (multivariate) polynomial coefficients in the desired recurrence. Thus, 0 means searching for recurrence equations with constant coefficients, 1 means that coefficients may be linear polynomials, and so on.

Individual degree bounds may be specified for each variable block. For doing so, instead of a single integer, put a list of integers whose length agrees with the number of blocks. Then the i th element in this list is used as bound for the total degree with respect to the variables in the i th block. Note that specifying a single integer d is *different* from specifying $\{d, d, d, ...\}$, because in the latter case, the degree counts individually for each variable, whereas in the former case, a total degree bound is applied. Thus, $n_1{}^d\, n_2{}^d\, n_3{}^d$ ... is covered by the latter, but not by the former.

■ Options

The specification of data array, structure set, variables, and degree bound are obligatory and immutable. After that parameter sequence, zero or more options can be given for fine tuning the behavior of the guesser. The possible options are described, item per item, in the following section.

■ Options

■ AdditionalEquations

The main computational step underlying the guesser is solving an overdetermined linear system of equations. Any solution of that system, if there is any, is considered as a reasonable recurrence equation which does not hold "for generic reasons" (because generically, an overdetermined linear system has no solution). There is some freedom in how overdetermined the system should be. This is governed by the **`AdditionalEquations`** options. By default, the value is set to 100, which is a good choice in most cases.

If the guesser produces unreasonable output (see below: Possible Issues), a standard advise is to use *more* additional equations. The maximum possible number of possible different equations is taken upon saying **`AdditionalEquations -> Infinity`**. (Caution: for multivariate examples, this may easily be hundreds of thousands of equations, more than will likely fit into your memory. Better increase carefully by hand, say to 500, then to 1000, then to 2000, etc.) If this still produces strange output, it is probably necessary to supply more data.

In time consuming computations, it may sometimes also be advantageous to use *less* additional equations and the default 100. The smallest possible value is 0, which often enough suffices to accurately produce the desired recurrence (corresponding to the linear algebra fact that a square matrix generically has full rank).

■ AdditionalTerms

This option is for fine tuning the terms allowed to appear in a recurrence: it specifies terms which may occur in addition to those given by structure set and degree bound. The terms given here consist of both the monomial part and the shift part: $n^\alpha\, f[n + \beta]$.

Example:

```
In[24]:= GuessMultRE[Table[1 / n!, {n, 0, 5}], {f[n], f[n + 1]}, {n}, 1]
Out[24]= {-f[n] + (1 + n) f[1 + n]}

In[25]:= GuessMultRE[Table[1 / n!, {n, 0, 5}], {f[n], f[n + 1]}, {n}, 0]
Out[25]= {}

In[26]:= GuessMultRE[Table[1 / n!, {n, 0, 5}], {f[n], f[n + 1]}, {n}, 0, AdditionalTerms → {n f[n + 1]}]
Out[26]= {-f[n] + (1 + n) f[1 + n]}
```

See also: Except

■ **Constraints**

Many bivariate sequences arising from combinatorics are nonzero only in a certain range, for instance in the triangle $0 \le k \le n$. Since the guessing command requires a rectangular array, the data array supplied will contain a lot of zeros, which may cause the guesser to slow down or, worse, to get confused about what recurrences are reasonable and which ones are just artefacts.

For such situations, it is possible to specify by the **Constraints** option the region of interest in the given data array. The region may be specified by (logical combinations of) polynomial inequalities, using the variables of the variable declaration for referring to the indices of the array.

Example: (The effect of this option is only noticable on larger examples.)

```
In[27]:= data = Table[Binomial[n, k], {n, 0, 5}, {k, 0, 5}];

In[28]:= GuessMultRE[data, {f[n, k], f[n, k + 1], f[n + 1, k], f[n + 1, k + 1]},
          {n, k}, 1, Constraints → 0 ≤ k ≤ n]

Out[28]= {-n f[n, k] - n f[n, 1 + k] + n f[1 + n, 1 + k],
          -n f[n, k] + f[n, 1 + k] + k f[1 + n, 1 + k], -f[n, k] - f[n, 1 + k] + f[1 + n, 1 + k],
          -(1 + n) f[n, k] - (-1 + k - n) f[1 + n, k], (k - n) f[n, k] + (1 + k) f[n, 1 + k]}
```

■ **Except**

This option is for fine tuning the terms allowed to appear in a recurrence: it specifies terms which may *not* occur in the recurrence, even though they are specified implicitly via structure set and degree bound. The terms given here must be given by both monomial part and shift part, i.e., in the form $n^{\alpha} f[n + \beta]$.

If a term in list of exceptional terms is enclosed in the keyword **Cone** then also all its multiples (both the monomial multiples and the shifted terms) will be excluded from consideration. This feature is useful for gradually guessing a Gröbner basis of the annihilating ideal.

Examples:

```
In[29]:= GuessMultRE[Table[HarmonicNumber[n], {n, 0, 15}], {f[n], f[n + 1], f[n + 2], f[n + 3]}, {n}, 2]

Out[29]= {(1 + n) (-5 + 2 n) f[n] - 3 (-3 - n + n^2) f[1 + n] + 5 f[2 + n] + (-3 + n) (3 + n) f[3 + n],
          2 (1 + n) f[n] - (4 + 3 n) f[1 + n] - f[2 + n] + (3 + n) f[3 + n],
          (-2 + n) (1 + n) f[n] - (-2 + n) (3 + 2 n) f[1 + n] + (-2 + n) (2 + n) f[2 + n],
          (1 + n) f[n] - (3 + 2 n) f[1 + n] + (2 + n) f[2 + n]}

In[30]:= GuessMultRE[Table[HarmonicNumber[n], {n, 0, 15}],
          {f[n], f[n + 1], f[n + 2], f[n + 3]}, {n}, 2, Except → {n f[n + 3]}]

Out[30]= {(1 + n) (-5 + 2 n) f[n] - 3 (-3 - n + n^2) f[1 + n] + 5 f[2 + n] + (-3 + n) (3 + n) f[3 + n],
          (-2 + n) (1 + n) f[n] - (-2 + n) (3 + 2 n) f[1 + n] + (-2 + n) (2 + n) f[2 + n],
          (1 + n) f[n] - (3 + 2 n) f[1 + n] + (2 + n) f[2 + n]}

In[31]:= GuessMultRE[Table[HarmonicNumber[n], {n, 0, 15}],
          {f[n], f[n + 1], f[n + 2], f[n + 3]}, {n}, 2, Except → {Cone[n f[n + 3]]}]

Out[31]= {(-2 + n) (1 + n) f[n] - (-2 + n) (3 + 2 n) f[1 + n] + (-2 + n) (2 + n) f[2 + n],
          (1 + n) f[n] - (3 + 2 n) f[1 + n] + (2 + n) f[2 + n]}
```

See also: AdditionalTerms

■ **Extension**

If the data is taken not from the rational numbers but from an algebraic number field, then this option allows for specifying the minimal polynomial of the primitive element. The data itself has then to be given in terms of the primitive element, for which some identifier has to be chosen (the same in the data and the minimal polynomial).

In[32]:=
```
GuessMultRE[Table[LegendreP[n, a], {n, 0, 15}],
 {f[n], f[n + 1], f[n + 2]}, {n}, 1, Extension → a^2 - 2]
```

Out[32]=
```
{(1 + n) f[n] - a (3 + 2 n) f[1 + n] + (2 + n) f[2 + n]}
```

- **Factor**

  If the guesser finds reasonable recurrences, it will factor their polynomial coefficients before presenting them to the user. In extremly big examples, chances are that this final factorization becomes the most time consuming part of the entire computation. In such situations it is advisable to switch of the factorization feature by specifying **`Factor -> False`** as option. (default is **`True`**.)

  Example

In[33]:=
```
GuessMultRE[Table[HarmonicNumber[2 n], {n, 0, 100}], {f[n], f[n + 1], f[n + 2]}, {n}, 3]
```

Out[33]=
$$\left\{\frac{1}{8}(1+n)(1+2n)(7+4n)f[n] - \frac{1}{8}(5+4n)\left(5+10n+4n^2\right)f[1+n] + \frac{1}{8}(2+n)(3+2n)(3+4n)f[2+n]\right\}$$

In[34]:=
```
GuessMultRE[Table[HarmonicNumber[2 n], {n, 0, 100}],
 {f[n], f[n + 1], f[n + 2]}, {n}, 3, Factor → False]
```

Out[34]=
$$\left\{\left(\frac{7}{8}+\frac{25n}{8}+\frac{13n^2}{4}+n^3\right)f[n] + \left(-\frac{25}{8}-\frac{35n}{4}-\frac{15n^2}{2}-2n^3\right)f[1+n] + \left(\frac{9}{4}+\frac{45n}{8}+\frac{17n^2}{4}+n^3\right)f[2+n]\right\}$$

- **Infolevel**

  The guesser can be made verbose by specifying a positive infolevel. The larger the infolevel, the more information about the ongoing computation will be printed on the screen. In big examples, this is useful for getting an idea if waiting longer makes sense or not.

In[35]:=
```
GuessMultRE[Table[HarmonicNumber[2 n], {n, 0, 100}],
 {f[n], f[n + 1], f[n + 2]}, {n}, 3, Infolevel → Infinity]
```

```
12 terms

collecting nonzero points...

modular system: 99 eqns, 12 vars

1 solutions predicted.

refined system: 99 eqns, 12 vars

Q.

2 147 483 629

2 147 483 587

{6.44016 × 10^-17, 0, 0.004, 1.28803 × 10^-16}

1 solutions.
```

Out[35]=
$$\left\{\frac{1}{8}(1+n)(1+2n)(7+4n)f[n] - \frac{1}{8}(5+4n)\left(5+10n+4n^2\right)f[1+n] + \frac{1}{8}(2+n)(3+2n)(3+4n)f[2+n]\right\}$$

Modulus

For reasons of efficiency, the guesser first solves the overdetermined linear system in a finite field. This allows to quickly get a prediction on the size of the solution space (and an early termination condition if it is empty), and it allows to determine beforehand which terms actually occur in the final answer and which don't. In many cases, this gives a quite considerable reduction for the linear system which is then, in a second step, solved over the rationals.

The option **Modulus** allows to specify the prime field which should be used for this preprocessing step. It will rarely be necessary to change the default prime 2147483629 to something else.

The preprocessing step can be skipped by specifying the modulus 0.

- **MustHaveOneOf**

This option allows to force terms into a recurrence: the guesser will output only those recurrences which contain at least one of the terms in the list specified by this option. The default is **All**, which does not impose any restriction.

Example.

```
In[36]:= data = Table[StirlingS1[n, k] Binomial[n, k], {n, 0, 20}, {k, 0, 20}];
         sset = Flatten[Table[f[n + i, k + j], {i, 0, 2}, {j, 0, 2}]];

In[38]:= GuessMultRE[data, sset, {n, k}, 3]

Out[38]= {-(k - n) (1 + n) f[n, k] - (1 + n) (3 + 3 k - 2 n + k n) f[n, 1 + k] - 3 (2 + k) n (1 + n) f[n, 2 + k] +
           (k - n) (3 + k + n) f[1 + n, 1 + k] + (2 + k) (5 + 3 k + n^2) f[1 + n, 2 + k] - (2 + k) (k - n) f[2 + n, 2 + k],
          -3 (k - n) (1 + n) f[n, k] - 3 (1 + k) n (1 + n) f[n, 1 + k] + (-1 + k - n) (2 + n) f[1 + n, k] +
           (1 + k) (2 + 3 k + n^2) f[1 + n, 1 + k] - (1 + k) (-1 + k - n) f[2 + n, 1 + k],
          (1 + k - n) (1 + n) f[n, 1 + k] + (2 + k) n (1 + n) f[n, 2 + k] - (2 + k) (1 + k - n) f[1 + n, 2 + k],
          (k - n) (1 + n) f[n, k] + (1 + k) n (1 + n) f[n, 1 + k] - (1 + k) (k - n) f[1 + n, 1 + k]}

In[39]:= GuessMultRE[data, sset, {n, k}, 3,
          MustHaveOneOf → {f[n + 2, k + 2], n f[n + 2, k + 2], k f[n + 2, k + 2]}]

Out[39]= {-1/2 (k - n) (1 + n) f[n, k] - 1/2 (1 + n) (3 + 3 k - 2 n + k n) f[n, 1 + k] -
           3/2 (2 + k) n (1 + n) f[n, 2 + k] + 1/2 (k - n) (3 + k + n) f[1 + n, 1 + k] +
           1/2 (2 + k) (5 + 3 k + n^2) f[1 + n, 2 + k] - 1/2 (2 + k) (k - n) f[2 + n, 2 + k]}
```

- **Return**

This option allows for specifying the return value of the guesser. By default, a list of recurrence equations is returned which forms a basis of the linear space of all recurrence of the requested type matching the data. Alternative outputs can be requested, by setting the **Return** option to an appropriate keyword. The possible choices are summarized in the following table.

"det"	Returns the determinant of the linear system (requires AdditionalEquations to be set to zero)
"dim"	Returns the dimension of the solution space according to the result obtained in the finite field
"except"	Returns the list of exceptional terms with Cone constructs expanded, intersected with the terms specified by structure set and degree bound
"mod"	Returns the modular solution (requires Modulus to be nonzero).
"sol"	Returns the solution (default)
"support"	Returns the list of terms actually appearing in the solution according to the finite field computation
"sys"	Returns the linear system as *a* matrix.
"terms"	Returns the list of terms to be taken into account

Example.

```
In[40]:= GuessMultRE[Table[2^k, {k, 0, 10}], {f[n], f[n + 1]}, {n}, 0, Return → "dim"]

Out[40]= 1

In[41]:= GuessMultRE[Table[2^k, {k, 0, 10}], {f[n], f[n + 1]}, {n}, 0, Return → "sys"]

Out[41]= {{1, 2}, {2, 4}, {4, 8}, {8, 16}, {16, 32},
          {32, 64}, {64, 128}, {128, 256}, {256, 512}, {512, 1024}}

In[42]:= NullSpace[%]

Out[42]= {{-2, 1}}

In[43]:= GuessMultRE[Table[2^k, {k, 0, 10}], {f[n], f[n + 1]}, {n}, 0, Return → "terms"]

Out[43]= {f[n], f[1 + n]}

In[44]:= %.First[%%]

Out[44]= -2 f[n] + f[1 + n]

In[45]:= GuessMultRE[Table[2^k, {k, 0, 10}], {f[n], f[n + 1]}, {n}, 0]

Out[45]= {-2 f[n] + f[1 + n]}
```

- **Sort**

Order internally used for sorting terms. The order influences the order of columns in the linear systems. In some situations, it may be desirable to sort the terms according to some term order rather than to the (somewhat arbitrary) natural order of Mathematica.

Example.

```
In[46]:= GuessMultRE[Table[Binomial[n, k], {n, 0, 10}, {k, 0, 10}],
          {f[n, k], f[n + 1, k], f[n, k + 1], f[n + 1, k + 1]}, {n, k}, 1, Sort → (OrderedQ[{#1, #2}] &)]

Out[46]= {-n f[n, k] - n f[n, 1 + k] + n f[1 + n, 1 + k],
          -n f[n, k] + f[n, 1 + k] + k f[1 + n, 1 + k], -f[n, k] - f[n, 1 + k] + f[1 + n, 1 + k],
          -(1 + n) f[n, k] - (-1 + k - n) f[1 + n, k], (k - n) f[n, k] + (1 + k) f[n, 1 + k]}

In[47]:= GuessMultRE[Table[Binomial[n, k], {n, 0, 10}, {k, 0, 10}],
          {f[n, k], f[n + 1, k], f[n, k + 1], f[n + 1, k + 1]}, {n, k}, 1, Sort → (OrderedQ[{#2, #1}] &)]

Out[47]= {f[n, k] - n f[n, 1 + k] - (1 + k - n) f[1 + n, 1 + k],
          k f[n, k] + k f[n, 1 + k] - k f[1 + n, 1 + k], n f[n, k] + n f[n, 1 + k] - n f[1 + n, 1 + k],
          (1 + n) f[n, 1 + k] + (k - n) f[1 + n, 1 + k], -(-1 + k - n) f[1 + n, k] - (1 + k) f[1 + n, 1 + k]}
```

- **StartPoint**

The guesser assumes by default that the array entry **data[[1,1,1,...]]** corresponds to the point $f[0, 0, 0, ...]$ of the sequence f under consideration. Occasionally, other start points may be desired. In this case, alternative start points can be declared by the start point option.

Example.

```
In[48]:= data = Table[HarmonicNumber[n], {n, 10, 150}];

In[49]:= GuessMultRE[data, {f[n], f[n + 1], f[n + 2]}, {n}, 1]

Out[49]= {(11 + n) f[n] - (23 + 2 n) f[1 + n] + (12 + n) f[2 + n]}
```

```
In[50]:= GuessMultRE[data, {f[n], f[n + 1], f[n + 2]}, {n}, 1, StartPoint → {10}]

Out[50]= {(1 + n) f[n] - (3 + 2 n) f[1 + n] + (2 + n) f[2 + n]}

In[51]:= data = Table[HarmonicNumber[n], {n, 0, 150}];

In[52]:= GuessMultRE[data, {f[n], f[n + 1], f[n + 2]}, {n}, 1]

Out[52]= {(1 + n) f[n] - (3 + 2 n) f[1 + n] + (2 + n) f[2 + n]}
```

■ Applications

■ Guessing an efficient evaluator

Consider the trivariate series expansion $1\Big/\Big(1 - x - y - z + \frac{3}{4}(x\,y + x\,z + y\,z)\Big) = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty}\sum_{k=0}^{\infty} c(n, m, k)\, z^k\, y^m\, x^n$. We seek a fast way for computing the coefficients $c(n, m, k)$. For small indices, we can use the series evaluator of Mathematica.

```
In[53]:= data = CoefficientList[Normal[
           Series[1 / (1 - x - y - z + 3 / 4 (x y + x z + y z)), {x, 0, 10}, {y, 0, 10}, {z, 0, 10}]], {x, y, z}];
```

Larger coefficients are more conveniently computed by a recurrence. Several recurrences are needed. First a recurrence in all three variables:

```
In[54]:= GuessMultRE[data, {c[n, m, k], c[n + 1, m, k], c[n, m + 1, k], c[n, m, k + 1]}, {n, m, k}, 1]

Out[54]= {-3/2 (2 + k + m + n) c[n, m, k] + (1 + k) c[n, m, 1 + k] + (1 + m) c[n, 1 + m, k] + (1 + n) c[1 + n, m, k]}
```

This recurrence is of first order, it determines all the coefficients $c(n, m, k)$ once the coefficients $c(0, m, k)$ are known. We next find a bivariate recurrence for those.

```
In[55]:= GuessMultRE[data[[1]], {c[m, k], c[m + 1, k], c[m, k + 1]}, {m, k}, 1]

Out[55]= {3/4 (k - m) c[m, k] - (1 + k) c[m, 1 + k] + (1 + m) c[1 + m, k]}
```

This recurrence is again of first order, and it allows the compuation of all the $c(0, m, k)$ once the coefficients $c(0, 0, k)$ are known. A recurrence for the latter can be spotted by the naked eye:

```
In[56]:= data[[1, 1]]

Out[56]= {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}
```

They are all one. Putting these results together and exploiting the symmetry of $c(n, m, k)$ in all arguments, we are led to the following procedure:

```
In[57]:= c[n_Integer, m_Integer, k_Integer] :=
          Which[
           ! OrderedQ[{n, m, k}], c @@ Sort[{n, m, k}],
           n == m == 0, 1,
           n == 0, c[0, m, k] = (k + 1) / m c[0, m - 1, k + 1] + 3 / 4 (-1 - k + m) / m c[0, m - 1, k],
           True, c[n, m, k] = (3 * (1 + k + m + n) * c[-1 + n, m, k]) / (2 * n) -
              ((1 + k) * c[-1 + n, m, 1 + k]) / n - ((1 + m) * c[-1 + n, 1 + m, k]) / n];
```

A quick cross check confirms the correctness of this procedure at least for small arguments:

In[58]:=
```
Table[c[n, m, k] - data[[n + 1, m + 1, k + 1]], {n, 0, 5}, {m, 0, 5}, {k, 0, 5}]
```

Out[58]=
```
{{{0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}}, {{0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
 {{0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}}, {{0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}},
 {{0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}}, {{0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}}}
```

The procedure using the recurrence equation is much faster than the direct computation using the series expansion. For example, it is a matter of just a few seconds to compute the first 100 diagonal elements:

In[59]:=
```
Timing[diag = Table[c[n, n, n], {n, 0, 100}];]
```

Out[59]=
```
{7.28846, Null}
```

If even more diagonal elements are desired, it is best to compute a special recurrence for the diagonal:

In[60]:=
```
GuessMultRE[diag, {d[n], d[n + 1], d[n + 2]}, {n}, 2]
```

Out[60]=
$$\left\{\frac{81}{128}\,(2+3\,n)\,(4+3\,n)\,d[n]-\frac{3}{16}\left(62+81\,n+27\,n^2\right)d[1+n]+(2+n)^2\,d[2+n]\right\}$$

Again, we turn this recurrence into a procedure:

In[61]:=
```
d[n_Integer] :=
  If[n < 10, c[n, n, n], d[n] = (-81 * (-4 + 3 * n) * (-2 + 3 * n) * d[-2 + n]) / (128 * n^2) +
     (3 * (8 - 27 * n + 27 * n^2) * d[-1 + n]) / (16 * n^2)];
```

This can do the first 100 terms even more quickly, and even the first 10000 terms can be found within a reasonalbe amount of time.

In[62]:=
```
Timing[diag = Table[d[n], {n, 0, 100}];]
```

Out[62]=
```
{0., Null}
```

In[63]:=
```
Timing[diag = Table[d[n], {n, 0, 10000}];]
```

Out[63]=
```
{2.52816, Null}
```

■ Guessing a Gröbner basis

The set of all recurrence equations satisfied by a fixed (multivariate) sequence can be considered as an ideal in the operator algebra $Q[n_1, n_2, \ldots][S_1, S_2, \ldots]$, where the n_i act on sequences by multiplication, and S_i by shift. Gröbner bases are preferred ideal bases. The **Except** option is useful for finding elements of a Gröbner basis: it allows to restrict the search to those terms which are "under the stairs" of the ideal generated by the recurrences already found. Let us illustrate this, without too much further comment, at the coefficients from the series expansion considered in the previous example. These coefficients, again, are computed by the following procedure:

In[64]:=
```
c[n_Integer, m_Integer, k_Integer] :=
  Which[
   ! OrderedQ[{n, m, k}], c @@ Sort[{n, m, k}],
   n == m == 0, 1,
   n == 0, c[0, m, k] = (k + 1) / m c[0, m - 1, k + 1] + 3 / 4 (-1 - k + m) / m c[0, m - 1, k],
   True, c[n, m, k] = (3 * (1 + k + m + n) * c[-1 + n, m, k]) / (2 * n) -
     ((1 + k) * c[-1 + n, m, 1 + k]) / n - ((1 + m) * c[-1 + n, 1 + m, k]) / n];
```

In[65]:=
```
data = Table[c[n, m, k], {n, 0, 50}, {m, 0, 50}, {k, 0, 50}];
```

The basic idea for finding a Gröbner basis is to apply the guesser to increasing sets of terms, excluding at each step all the terms which would just yield consequences of the recurrence equations already found in earlier iterations. Proceedings iteratively has two advantages: (1) the output is less redundand than with a direct computation, (2) the linear systems stay much smaller and thus the overall performance and the memory requirements are much more managable than with a direct computation. (This is in particular relevant for big examples.) Below, we execute the procedure for finding a Gröbner basis of the ideal generated by all recurrence equations for $c(n, m, k)$ of order four or less and degree four our less. For convenience, we have chosen a lexicographic term order with $f[n+1, m, k] > f[n, m+1, k] > f[n, m, k+1] > n\, f[n, m, k] > m\, f[n, m, k] > k\, f[n, m, k]$.

In[66]:=
```
basis = {}; lt = {};
```

In[67]:=
```
new = GuessMultRE[data,
  Flatten[Table[f[n + a, m + b, k + c], {a, 0, 1}, {b, 0, 1}, {c, 0, 1}]], {n, m, k}, 0]
```

Out[67]=
$$\left\{\frac{3}{4}\,f[n, m, 1+k] + \frac{3}{4}\,f[n, 1+m, k] - f[n, 1+m, 1+k] + \frac{3}{4}\,f[1+n, m, k] - f[1+n, m, 1+k] - f[1+n, 1+m, k] + f[1+n, 1+m, 1+k]\right\}$$

In[68]:=
```
basis = Union[basis, new]; lt = Union[lt, {Cone[f[n, m + 1, k + 1]]}];
```

In[69]:=
```
new = GuessMultRE[data,
  Flatten[Table[f[n + a, m + b, k + c], {a, 0, 1}, {b, 0, 1}, {c, 0, 1}]], {n, m, k}, 1, Except → lt]
```

Out[69]=
$$\left\{\frac{3}{4}\,(1+k)\,f[n, m, 1+k] - \frac{3}{4}\,(1+m)\,f[n, 1+m, k] + \frac{3}{4}\,(k-m)\,f[1+n, m, k] - (1+k)\,f[1+n, m, 1+k] + (1+m)\,f[1+n, 1+m, k],\ -\frac{3}{2}\,(2+k+m+n)\,f[n, m, k] + (1+k)\,f[n, m, 1+k] + (1+m)\,f[n, 1+m, k] + (1+n)\,f[1+n, m, k]\right\}$$

In[70]:=
```
basis = Union[basis, new]; lt = Union[lt, {Cone[m f[n + 1, m + 1, k]], Cone[n f[n + 1, m, k]]}];
```

In[71]:=
```
new = GuessMultRE[data,
  Flatten[Table[f[n + a, m + b, k + c], {a, 0, 1}, {b, 0, 1}, {c, 0, 1}]], {n, m, k}, 2, Except → lt]
```

Out[71]= {}

In[72]:=
```
new = GuessMultRE[data,
  Flatten[Table[f[n + a, m + b, k + c], {a, 0, 1}, {b, 0, 1}, {c, 0, 1}]], {n, m, k}, 3, Except → lt]
```

Out[72]= {}

In[73]:=
```
new = GuessMultRE[data,
  Flatten[Table[f[n + a, m + b, k + c], {a, 0, 1}, {b, 0, 1}, {c, 0, 1}]], {n, m, k}, 4, Except → lt]
```

Out[73]= {}

In[74]:=
```
new = GuessMultRE[data,
  Flatten[Table[f[n + a, m + b, k + c], {a, 0, 2}, {b, 0, 2}, {c, 0, 2}]], {n, m, k}, 1, Except → lt]
```

Out[74]=
$$\left\{\frac{63}{256}(2+k+m+n)\,f[n,m,k]+\frac{3}{128}(15+11k+11m-7n)\,f[n,m,1+k]-\frac{11}{32}(2+k)\,f[n,m,2+k]-\right.$$
$$\frac{21}{64}(1+m)\,f[n,1+m,k]+\frac{3}{4}(1+k)\,f[1+n,m,k]-\frac{1}{16}(17+13k-5m)\,f[1+n,m,1+k]+$$
$$\frac{3}{4}(1+k)\,f[2+n,m,k]-\frac{1}{4}(11+7k+m)\,f[2+n,m,1+k]+(2+k)\,f[2+n,m,2+k],$$
$$\frac{9}{64}(2+k+m+n)\,f[n,m,k]+\frac{3}{32}(9+5k+5m-n)\,f[n,m,1+k]-$$
$$\frac{5}{8}(2+k)\,f[n,m,2+k]-\frac{3}{16}(1+m)\,f[n,1+m,k]+\frac{3}{4}(1+k)\,f[1+n,m,k]-$$
$$\left.\frac{1}{4}(11+7k+m)\,f[1+n,m,1+k]+(2+k)\,f[1+n,m,2+k]\right\}$$

In[75]:=
```
basis = Union[basis, new]; lt = Union[lt, {Cone[k f[n + 1, m, k + 2]]}];
```

In[76]:=
```
new = GuessMultRE[data,
  Flatten[Table[f[n + a, m + b, k + c], {a, 0, 2}, {b, 0, 2}, {c, 0, 2}]], {n, m, k}, 2, Except → lt]
```

Out[76]=
$$\left\{-\frac{9}{8}(k-m)\,(2+k+m+n)\,f[n,m,k]+\frac{3}{4}(1+k)\,(5+3k+m+n)\,f[n,m,1+k]-\right.$$
$$\left.(1+k)\,(2+k)\,f[n,m,2+k]-\frac{3}{4}(1+m)\,(5+k+3m+n)\,f[n,1+m,k]+(1+m)\,(2+m)\,f[n,2+m,k]\right\}$$

In[77]:=
```
basis = Union[basis, new]; lt = Union[lt, {Cone[m^2 f[n, m + 2, k]]}];
```

In[78]:=
```
new = GuessMultRE[data,
  Flatten[Table[f[n + a, m + b, k + c], {a, 0, 2}, {b, 0, 2}, {c, 0, 2}]], {n, m, k}, 3, Except → lt]
```

Out[78]= {}

In[79]:=
```
new = GuessMultRE[data,
  Flatten[Table[f[n + a, m + b, k + c], {a, 0, 2}, {b, 0, 2}, {c, 0, 2}]], {n, m, k}, 4, Except → lt]
```

Out[79]= {}

In[80]:=
```
new = GuessMultRE[data,
  Flatten[Table[f[n + a, m + b, k + c], {a, 0, 3}, {b, 0, 3}, {c, 0, 3}]], {n, m, k}, 1, Except → lt]
```

Out[80]= {}

In[81]:=
```
new = GuessMultRE[data,
  Flatten[Table[f[n + a, m + b, k + c], {a, 0, 3}, {b, 0, 3}, {c, 0, 3}]], {n, m, k}, 2, Except → lt]
```

Out[81]=
$$\left\{-\frac{3}{4}(4+4k-m+n)\,(2+k+m+n)\,f[n,m,k]+\right.$$
$$\frac{1}{2}\left(60+66k+18k^2+17m+11km+m^2+19n+13kn-2mn+n^2\right)f[n,m,1+k]-$$
$$\frac{2}{3}(2+k)\,(30+13k+4m+4n)\,f[n,m,2+k]+$$
$$\left.\frac{8}{3}(2+k)\,(3+k)\,f[n,m,3+k]-(1+m)\,(m-n)\,f[n,1+m,k]\right\}$$

In[82]:=
```
basis = Union[basis, new]; lt = Union[lt, {Cone[n m f[n, m + 1, k]]}];
```

```
In[83]:= new = GuessMultRE[data,
           Flatten[Table[f[n + a, m + b, k + c], {a, 0, 3}, {b, 0, 3}, {c, 0, 3}]], {n, m, k}, 3, Except → lt]
Out[83]= {}

In[84]:= new = GuessMultRE[data,
           Flatten[Table[f[n + a, m + b, k + c], {a, 0, 3}, {b, 0, 3}, {c, 0, 3}]], {n, m, k}, 4, Except → lt]
Out[84]= {}

In[85]:= new = GuessMultRE[data,
           Flatten[Table[f[n + a, m + b, k + c], {a, 0, 4}, {b, 0, 4}, {c, 0, 4}]], {n, m, k}, 1, Except → lt]
Out[85]= {}

In[86]:= new = GuessMultRE[data,
           Flatten[Table[f[n + a, m + b, k + c], {a, 0, 4}, {b, 0, 4}, {c, 0, 4}]], {n, m, k}, 2, Except → lt]
Out[86]= {27/32 (1 + k) (2 + k + m + n) f[n, m, k] -
           9/32 (54 + 53 k + 13 k^2 + 17 m + 10 k m + m^2 + 17 n + 10 k n - 2 m n + n^2) f[n, m, 1 + k] +
           3/16 (204 + 158 k + 31 k^2 + 38 m + 16 k m + m^2 + 38 n + 16 k n - 2 m n + n^2) f[n, m, 2 + k] -
           1/4 (3 + k) (49 + 16 k + 4 m + 4 n) f[n, m, 3 + k] + (3 + k) (4 + k) f[n, m, 4 + k]}

In[87]:= basis = Union[basis, new]; lt = Union[lt, {Cone[k^2 f[n, m, k + 4]]}];

In[88]:= new = GuessMultRE[data,
           Flatten[Table[f[n + a, m + b, k + c], {a, 0, 4}, {b, 0, 4}, {c, 0, 4}]], {n, m, k}, 3, Except → lt]
Out[88]= {}

In[89]:= new = GuessMultRE[data,
           Flatten[Table[f[n + a, m + b, k + c], {a, 0, 4}, {b, 0, 4}, {c, 0, 4}]], {n, m, k}, 4, Except → lt]
Out[89]= {}
```

At this point, **basis** carries the desired Gröbner basis. The leading terms are as follows:

```
In[90]:= lt
Out[90]= {Cone[k^2 f[n, m, 4 + k]], Cone[m n f[n, 1 + m, k]], Cone[f[n, 1 + m, 1 + k]], Cone[m^2 f[n, 2 + m, k]],
          Cone[n f[1 + n, m, k]], Cone[k f[1 + n, m, 2 + k]], Cone[m f[1 + n, 1 + m, k]]}
```

■ Possible Issues

■ Unreasonable or varying predictions

The **Infolevel** option is especially useful for large examples, as it prints out a reasonable prediction on the number of solutions before the main part of the computation. If the number of predicted solutions is large, it is a good idea to abort the computation and restart with a smaller degree and/or a smaller structure set. This does not only result in an improved efficiency, but is also influences the probablility that artefact solutions appear: the larger the solution set, the more likely it becomes that some additional unwanted solutions appear as well. This phenomenon, which typically arises only on very large input with some degenerate properties, can be observed by running the prediction (**Return->"dim"**) several times: if if does not always produce the same number, this is a good indication that something is wrong. In this case, try increasing the **AdditionalEquations** and/or the amount of input data, or reduce the degree bounds and structure set.

Unreasonably long coefficients in the output

Recurrence equations of of high order with high degree may easily involve integers with several dozens of decimal digits. This need not be a sign that the recurrence is incorrect. However, artefact recurrences virtually *always* involve extremely long integers and have dense irreducible polynomials as coefficients. If, therefore, the guesser returns a set of recurrences where one of them is much more ugly than the others, then the ugly recurrence might be an artefact. In this case, increase **`AdditionalEquations`**, reduce degree bounds and structure set, and provide more data. Artefact recurrence equations can also be detected by running the guesser several times: the valid recurrences are likely to always be the same, while artefact outputs tend to change from one computation to the other. Also the number may differ.

- **Unreasonably long compuation time**

The computation time depends most dominaltly of the number of terms allowed to appear in the recurrence, but it also depends on the length of the integers appearing in the final output. The latter relation is approximately linear, so that it can be rougly expected that the one computation will take about twice as long as a second computation (with the same set of allowed terms) if the integers appearing in the result of the first computation are about twice as long as the integers appearing in the result of the second. Now, since artefact recurrence equations tend to involve extremly long integers, their appearence in the output slows down the computation dramatically. Therefore, if a computation is not completed within a reasonalbe amount of time, this may be seen as an indication that there are artefact results being computed. In this event, try increasing the **`AdditionalEquations`** and/or the amount of input data, or reduce the degree bounds and the structure set. Observe the computation with at an appropriate **`Infolevel`**.

- **Many zeros in the data array**

Continous ranges of zero in the data array do not contribution any information about the recurrence equations to be deter–mined. Lots of zeros also increase the probablity of getting artefacts in the result. Therefore, if the guesser detects a suspicious percentage of zeros in the data array, it will sort them out before starting the main computation. In some cases, especially in higher dimensional data arrays, this step may take a considerable amount of time. With higher values of **`Infolevel`**, warnings are issued if special actions to avoid problems resulting from zero ranges in the input array. If the computation gets stuck after such a warning, it is strongly recommended to use the **`Constraints`** option for restricting the guesser's attention to the interesting parameter space, i.e., the area in which there no (or just a few) zeros.

Univariate Guessing

GuessMinRE: Minimal Recurrence Equations

We provide a special purpose guesser for univariate recurrence equations. This guesser does not require terms or degrees to be specified, but instead searches for the (reasonable) recurrence of minimal possible order matching the given data. This recurrence is not only unusally the most interesting recurrence for the problem at hand, but it is also in many cases (in particular in large examples) the recurrence with least possible integers appearing in it, and therefore it is the one which can be most efficiently computed. The command accepts just a list of rational numbers and a function symbol **`f[n]`** specifying the symbols **`f`** and **`n`** to be used in the output. If no recurrence can be found, an error is thrown.

```
In[91]:= GuessMinRE[{1, 1, 2, 3, 5, 8, 13, 21, 35}, f[n]]

Out[91]= -f[n] - f[1 + n] + f[2 + n]
```

```
In[92]:= GuessMinRE[Table[HarmonicNumber[n]^2 + HarmonicNumber[2 n] + 1 / n!, {n, 0, 60}], f[n]]
```

$$\text{Out[92]}= \left(-\frac{483}{8}-\frac{10\,839\,n}{32}-\frac{5447\,n^2}{8}-\frac{3603\,n^3}{8}+\frac{1689\,n^4}{4}+\frac{8321\,n^5}{8}+\frac{14\,227\,n^6}{16}+\frac{13\,313\,n^7}{32}+\frac{449\,n^8}{4}+\frac{131\,n^9}{8}+n^{10}\right) f[n]+\left(\frac{4317}{8}+\frac{103\,983\,n}{32}+\frac{60\,439\,n^2}{8}+\frac{266\,641\,n^3}{32}+\frac{103\,841\,n^4}{32}-\frac{79\,869\,n^5}{32}-\frac{128\,567\,n^6}{32}-\frac{39\,445\,n^7}{16}-\frac{27\,501\,n^8}{32}-\frac{711\,n^9}{4}-\frac{163\,n^{10}}{8}-n^{11}\right) f[1+n]+\left(-\frac{5517}{4}-\frac{286\,299\,n}{32}-\frac{369\,451\,n^2}{16}-\frac{959\,581\,n^3}{32}-\frac{605\,969\,n^4}{32}-\frac{42\,907\,n^5}{32}+\frac{220\,879\,n^6}{32}+\frac{88\,027\,n^7}{16}+\frac{68\,763\,n^8}{32}+\frac{957\,n^9}{2}+\frac{465\,n^{10}}{8}+3\,n^{11}\right) f[2+n]+\left(1383+\frac{306\,435\,n}{32}+\frac{213\,421\,n^2}{8}+\frac{1\,202\,789\,n^3}{32}+\frac{866\,923\,n^4}{32}+\frac{198\,007\,n^5}{32}-\frac{171\,529\,n^6}{32}-\frac{85\,111\,n^7}{16}-\frac{69\,767\,n^8}{32}-\frac{1961\,n^9}{4}-\frac{473\,n^{10}}{8}-3\,n^{11}\right) f[3+n]+\left(-483-3540\,n-\frac{167\,375\,n^2}{16}-\frac{495\,437\,n^3}{32}-\frac{378\,307\,n^4}{32}-\frac{108\,515\,n^5}{32}+\frac{50\,763\,n^6}{32}+\frac{59\,745\,n^7}{32}+\frac{24\,913\,n^8}{32}+\frac{1385\,n^9}{8}+\frac{163\,n^{10}}{8}+n^{11}\right) f[4+n]$$

The command accepts the options **Modulus**, **Inputlevel**, and **StartPoint**, whose semantics is similar as described above for **GuessMultRE**. There is also an option **Return**, which may carry the values **"rec"** (default; for returning the recurrence) or **"bool"** (for returning just **True** or **False** depending on whether or not a recurrence was found).

Note that **GuessMinRE** is able to discover recurrences which **GuessMultRE** would only be able to find if more data is provided:

```
In[93]:= GuessMultRE[Table[HarmonicNumber[n]^2 + HarmonicNumber[2 n] + 1 / n!, {n, 0, 60}],
          {f[n], f[n + 1], f[n + 2], f[n + 3], f[n + 4]}, {n}, 11]

Throw::nocatch : Uncaught Throw[insufficient input data.] returned to top level. >>

Out[93]= Hold[Throw[insufficient input data.]]
```

In cases where both functions apply, **GuessMinRE** is typically much faster than **GuessMultRE**.

■ GuessMinDE: Minimal Differential Equations

This is like **GuessMinRE**, but it returns a differential equation for the generating function rather than a recurrence equation for the coefficients themselves. For example,

```
In[94]:= GuessMinDE[Table[HarmonicNumber[n], {n, 0, 25}], f[x]]

Out[94]= f[x] + (-3 + 3 x) f'[x] + (1 - 2 x + x^2) f''[x]

In[95]:= % /. f → (1 / (# - 1) Log[1 - #] &) // FullSimplify

Out[95]= 0
```

The same options as for **GuessMinRE** apply.

GuessMinAE: Minimal Algebraic Equations

With this command, it is possible to find algebraic equations, given the first coefficients in the series expansion of an algebraic function. For example,

```
In[96]:= GuessMinAE[Table[Binomial[1 / 2, n], {n, 0, 10}], f[x]]
```

$$\text{Out[96]}= 1 + x - f[x]^2$$

```
In[97]:= GuessMinAE[Table[16^n Pochhammer[5 / 6, n]
    Pochhammer[1 / 2, n] / Pochhammer[5 / 3, n] / Pochhammer[2, n], {n, 0, 100}], f[x]]
```

$$\text{Out[97]}= -\frac{1}{27} + \frac{16\,x}{9} - \frac{64\,x^2}{3} - \frac{256\,x^3}{27} + \left(\frac{1}{27} - \frac{20\,x}{9} + \frac{304\,x^2}{9} - \frac{512\,x^3}{27}\right) f[x] + \left(\frac{10\,x}{27} - \frac{104\,x^2}{9} + \frac{208\,x^3}{9} - \frac{512\,x^4}{27}\right) f[x]^2 + \left(\frac{5\,x^2}{3} - \frac{56\,x^3}{3} - \frac{64\,x^4}{3}\right) f[x]^3 + \left(\frac{13\,x^3}{3} - \frac{28\,x^4}{3} - \frac{32\,x^5}{3}\right) f[x]^4 + 7\,x^4\,f[x]^5 + 7\,x^5\,f[x]^6 + 4\,x^6\,f[x]^7 + x^7\,f[x]^8$$

Technical Issues

Version

The package was developped for Mathematica 5.2 and adpated to run as well for Mathematica 6 and Mathematica 7. The version of the package can be queried via

```
In[98]:= Guess`Private`Version
```

```
Out[98]= 0.32 2009-04-07
```

The latest version of this software can be found on http://www.risc.uni−linz.ac.at/research/combinat/software/

Installation

The package is installed by simply putting the package file to a place where Mathematica is able to find it.

Licence

This software is free; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. The programs are distributed in the hope that they will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. See the GNU General Public License (http://www.gnu.org/licenses/gpl.html) for more details.

Bugs

Please report bugs to Manuel Kauers (mkauers@risc.uni−linz.ac.at)

Bibliography

This software has already proven useful. It has been applied in the recent works described in the following articles.

[1] Johannes Blümlein, Manuel Kauers, Carsten Schneider, and Sebastian Klein. *From Moments to Functions in Quantum Chromodynamics.* ArXiv:0902.4095, 2009. (submitted)

[2] Johannes Blümlein, Manuel Kauers, Carsten Schneider, and Sebastian Klein. *Determining the closed forms of the $O(a_s^3)$ anomalous dimension and Wilson coefficients from Mellin moments by means of computer algebra.* ArXiv:0902.4091. (submitted)

[3] Alin Bostan and Manuel Kauers. *The full generating function of Gessel walks is algebraic.* 2009 (in preparation)

[4] Alin Bostan and Manuel Kauers. *Automated classification of restricted lattice walks.* In proceedings of FPSAC'09. (to appear)

[5] Manuel Kauers. *Computer algebra and power series with postive coefficients.* In proceedings of FPSAC'07.

[6] Manuel Kauers, Christoph Koutschan, and Doron Zeilberger. *Proof of Ira Gessel's lattice path conjecture.* 2009 (submitted)

[7] Manuel Kauers, and Christoph Koutschan, and Doron Zeilberger. *A proof of George Andrews' and Dave Robbins' q−TSPP−conjecture (modulo a finite amount of routine calculations).* The personal Journal of Shalosh B. Eckhad and Doron Zeilberger, also ArXiv:0808.0571, 2009.

The following are other software packages providing guessing functionality

[8] Christian Mallinger. *Algorithmic Manipulations and Transformations of Univariate Holonomic Functions and Sequences.* Master's thesis. RISC−Linz. 1996

[9] Bruno Salvy and Paul Zimmermann. *Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable.* ACM Transactions on mathematical software. 20(2):163−−177, 1994.

[10] Christian Krattenthaler. *RATE: A Mathematica Guessing Machine.* http://mat.univie.ac.at/~kratt/rate/rate.html

[11] Martin Ruby. *Extended Rate, more GFUN.* ArXiv:0702086, 2007.