Interpolation in Symbolic Summation

Fabrizio Caruso
Dipartimento di Matematica "Tonelli", University of Pisa, Italy
email: caruso@dm.unipi.it

Introduction

We present two software packages for computing the null space of a matrix with rational function coefficients. We use two approaches based on classical interpolation algorithms. The packages are written in the Mathematica language.
In the first package we use polynomial interpolation and Cramer's rule to reconstruct the components of the solutions. This package works under some assumptions on the matrix that, in most cases, are true for systems involved in Zeilberger's algorithm for hypergeometric definite summation.
In the second package we use a hybrid method based on heuristic rational function interpolation on some low degree components of the solutions and "symbolic" Gaussian elimination on the other components. The second package works under very general conditions and can be customized for systems related to symbolic summation.

Keywords : Hypergeometric Summation, Interpolation, Zeilberger's algorithm, Mathematica

Solving by Polynomial Interpolation and Cramer's Rule

We use Cramer's rule and polynomial interpolation to solve overdetermined systems of homogeneous linear equations over $\mathbb{Q}$ and over the field of rational functions over $\mathbb{Q}$ under the hypothesis that the null space has dimension 1 and that the system is overdetermined. These conditions very often hold in the systems involved in Zeilberger's algorithm for definite hypergeometric summation [Zei90], [PS95]. The algorithm described is a generalization of the algorithm in [Lip81].

The main steps of the algorithm are

1. We rewrite the homogeneous system as an equivalent inhomogeneous system
2. We solve sufficiently many homomorphic images of the problem
3. We reconstruct the solutions by Cramer's rule and polynomial interpolation

The Homogeneous Problem We consider the problem of solving the homogeneous system of linear equations

$$A\mathbf{x} = 0, \tag{1}$$

with $A \in \mathbb{K}^{m,n}$, $m \geq n$, where $\mathbb{K}$ is either $\mathbb{Q}$ or $\mathbb{Q}(y)$ and where the dimension of the null space of A is 1.
Let us denote the matrix obtained from A by removing the last column by A' and let us assume that A' has maximal rank, i. e. $n-1$.
We consider the problem:

$$A'\mathbf{x} = b, \tag{2}$$

where b is a column vector obtained from the last column of A obtained by taking the components with the opposite sign.

Solving the Homomorphic Problems We can solve the overdetermined system of linear equations of maximal rank (2) by computing several homomorphic images of the "Cramer determinants" (by modular computation for $\mathbb{K} = \mathbb{Q}$ and by evaluation for $\mathbb{K} = \mathbb{Q}(y)$).
More precisely:

1. each homomorphic system is triangulated by Gaussian elimination,
2. the determinant d of the upper square part of the triangulated matrix, which is the last component ($d = x_n$) of the solution to (1), is computed,
3. the solutions to the inhomogeneous problem (2) $s_1, s_2, \ldots, s_{n-1}$ are computed,
4. the numerators of the solutions to (2), which are the remaining components of the solution to (1), are obtained by Cramer's rule, i. e. $x_i = s_i \cdot d$, for $i = 1, \ldots, n-1$.

A more detailed description is in [Car03].

Bounding the Solutions Unless a good bound for the absolute value of "Cramer determinants" for $\mathbb{K} = \mathbb{Q}$ is known because of extra information about the specific problem, a reasonable bound is given by the *Hadamard bound*.

Definition. For a given matrix $A \in \mathbb{Q}^{n,n}$, the *Hadamard bound*, denoted by $H(A)$, for the absolute value of the determinant of A is given by

$$H(A) = \prod_{i=1}^{n} ||A_i||_2, \tag{3}$$

where A_i is the i-th row in A.

Bounding the Degrees of the Solutions In order to have a bound on the degree of the "Cramer determinants" for $\mathbb{K} = \mathbb{Q}(y)$ we can again use *Hadamard bound*.
In fact, for any square matrix $M \in \mathbb{Q}[y]^{n,n}$ we have that

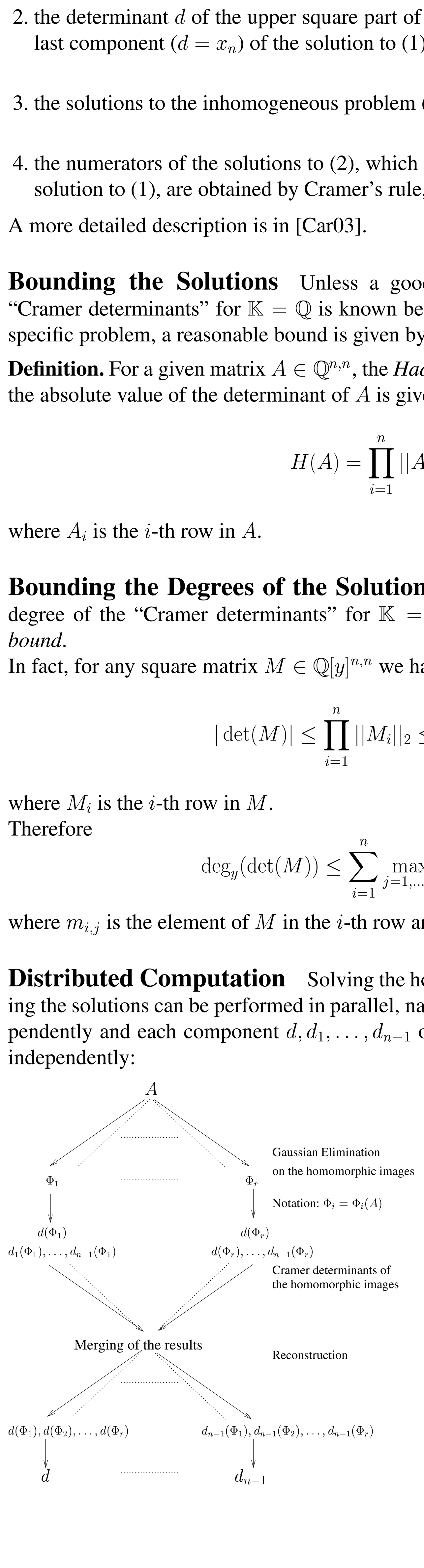
$$|\det(M)| \leq \prod_{i=1}^{n} ||M_i||_2 \leq \prod_{i=1}^{n} ||M_i||_1; \tag{4}$$

where M_i is the i-th row in M.
Therefore

$$\deg_y(\det(M)) \leq \sum_{i=1}^{n} \max_{j=1,\ldots,n} (\deg_y(m_{i,j})), \tag{5}$$

where $m_{i,j}$ is the element of M in the i-th row and j-th column.

Distributed Computation Solving the homomorphic images and reconstructing the solutions can be performed in parallel, namely each image can be solved independently and each component $d, d_1, \ldots, d_{n-1}$ of the solutions can be reconstructed independently:



These algorithms are implemented in a Mathematica package. The parallel version uses the Java library "Distributed Mathematica" [PS00].
The package can be obtained from the author by sending an email to caruso@dm.unipi.it.

The Commands The package `CramerNullSpace.m` provides the following commands for the case $\mathbb{K} = \mathbb{Q}(y)$:

`ChNullSpaceAt`	null space for a given value of y
`ChNullSpaceUntil`	null space by interpolation in a given number of points
`DistChNullSpaceUntil`	null space in parallel by interpolation in a given number of points

An Example We can use `ChNullSpaceUntil` or `DistChNullSpaceUntil` to find the null space of the following matrix by using 10 evaluations

$$A = \begin{pmatrix} 1 & 0 & 0 & 2n+1 \\ -2n-2 & 0 & 2n+2 & -n^2+1 \\ n^2+2n+1 & n^2+2n+1 & -n^2-2n-1 & -n^2-2n-1 \end{pmatrix} \tag{6}$$

as follows

```
ChNullSpaceUntil[A, 10] // Simplify
```
$\{\{2(1+n)^3(1+2n), -(1+n)^4, 3(1+n)^4, -2(1+n)^3\}\}$
We can do the same but in parallel with `DistChNullSpaceUntil`:

```
<< dist.m
```
Distributed Mathematica V1.0.0 (c) 2000 Cleopatra Pau (RISC-Linz)
See http://www.risc.uni-linz.ac.at/software/distmath
```
InitializeD[pinwheel, octane, andromeda, octane, galaxy, solaris]
```
Connecting pinwheel...
Connecting andromeda...
Connecting galaxy...
okay
```
AllD["<< CramerNullSpace.m"]
```
okay
`AllD["myCoefs =` $\{\{1,0,0,1+2n\},\{-2-2n,0,2+2n,1-n^2\},\{1+2n+n^2,1+2n+n^2,-1-2n-n^2,-1-2n-n^2\}\}$`"]`
okay
```
DistChNullSpaceUntil[myCoefs, 10] // Simplify
```
$\{\{2(1+n)^3(1+2n), -(1+n)^4, 3(1+n)^4, -2(1+n)^3\}\}$

Remark. The command `InitializeD` initializes the computers available on the network. The command `AllD` "broadcast" a command to all the computers in the network.

Timings This approach has proven to be inadequate mainly because of the excessive number of homomorphic images that must be computed. For this reason this package has not be further developed.
Here we present some timings in seconds of the built-in Mathematica command and `ChNullSpaceeUntil` for linear systems that are to be solved when using Zeilberger's fast algorithm for symbolic summation of powers of binomials, i. e. the problem of finding a linear recurrence with polynomial coefficients in n for $\sum_{k=0}^{n} \binom{n}{k}^p$.

power	built-in	`ChNullSpaceUntil`	hom. images
1	< 0.01	0.01	2
2	< 0.01	0.07	4
3	0.07	1.47	16

These timings were obtained with Mathematica 4.0 under Linux on a Pentium II at 400Mhz

Conclusion This approach performs very badly when compared with an approach based on rational interpolation.
The problems encountered are:

- The method works under restrictive hypotheses
- The method requires the detection of unlucky cases
- Too many homomorphic images are computed

The excessive number of necessary homomorphic images is caused by the fact that the "Cramer determinants" are the denominator and numerators of the components of the solution which very often have common factors that are not predicted in advance.

Solving by Heuristics and Rational Function Interpolation

We use rational function interpolation to build an algorithm that heuristically computes the null space of a rectangular matrix with entries in $\mathbb{Q}(y_1, \ldots, y_n)$ by reconstructing some low degree components of the solutions by rational function interpolation and the others symbolically. This is done by plugging the components reconstructed by rational function interpolation into the system in order to simplify it.

The Theory of Rational Function Interpolation For a detailed description of rational function interpolation we refer to [GvzG99]. In the following let F be a field, $m \in F[x]$ be a polynomial of positive degree n (which will serve as modulus), and $f \in F[x]$ be a polynomial of degree less than n.
In the following let $u_0, u_1, \ldots, u_{n-1}$ be distinct elements of F, and for $0 \leq i < n$, let $v_i = g(u_i)$ be a collection of samples of an unknown function $g : F \to F$.
Given $k \in \{0, 1, \ldots, n\}$, we want to find a rational function $r/t \in F(x)$, with $r, t \in F[x]$, such that

$$\begin{aligned} & t(u_i) \neq 0; \\ & r(u_i)/t(u_i) = v_i; \quad 0 \leq i < n, \ \deg r < k, \ \deg t \leq n-k. \end{aligned} \tag{7}$$

Theorem. *Let $f \in F[x]$ of degree less than n, with $f(u_i) = v_i$ for $0 \leq i < n$, let $k \in \{0, 1, \ldots, n\}$ and $r_j, s_j, t_j \in F[x]$ be the remainder and cofactors in the extended Euclidean algorithm for $((x-u_0)\cdot(x-u_1)\cdots(x-u_{n-1}), f)$, where j is minimal such that $\deg r_j < k$.*

1. *If $\gcd(r_j, t_j) = 1$ then r_j and t_j solve (7).*
2. *If $r/t \in F(x)$ is a canonical form solution to (7), then $\gcd(r_j, t_j) = 1$ and $r = \tau^{-1} r_j$ and $t = \tau^{-1} t_j$, where τ is the leading coefficient of t_j.*

Homogeneous Linear Systems We apply rational function interpolation to reconstruct the components of the solutions of a homogeneous system of linear equations with rational function coefficients.
Given $A \in \mathbb{Q}(y_1, \ldots, y_p)^{m,n}$, we want to find $\mathbf{x} = (x_1, \ldots, x_n)$, with $x_i \in \mathbb{Q}(y_1, \ldots, y_p)$, $1 \leq i \leq n$ such that

$$A\mathbf{x} = 0. \tag{8}$$

A detailed description of the methods used is in [Car03].

The Heuristic Method We use rational function interpolation in an iterative way by discarding those solutions that change once a new evaluation is considered and by keeping the others. Once we have a unique candidate, we take that as solution.

The Recursive Method Such improved heuristic reconstructs those components that require few evaluations to stabilize, then plugs their values into the system in order to simplify it and recursively reconstructs the remaining components in the simplified system.

The Hybrid Method An ever faster method is to use the heuristic method only for those components that require few evaluations to stabilize. Once these components have been computed their values can be plugged into the original system in order to simplify it. Then the simplified system can be solved symbolically. This method can be sped up by putting into the system extra information on the degree of the components of the solutions.
This method generalizes the work discussed in [RZ], where only zero components of the solutions are guessed by computing just one homomorphic image.

The Package A package that implements null space computation by rational function interpolation with different strategies (*heuristic*, *recursive*, *hybrid method*) has been implemented in the Mathematica computer algebra system.
The package can be obtained from the author by sending an email to caruso@dm.unipi.it.

The Commands Null space computation is implemented in the functions

`ModularNullSpace`	null space (heuristic method)
`MultiModularNullSpace`	multivariate null space (heuristic method)
`HybridNullSpace`	null space (hybrid/recursive method)
`MultiHybridNullSpace`	multivariate null space (hybrid/recursive method)

These commands take as input a matrix and output the null space. Both input and output follow the Mathematica convention used in the internal `NullSpace` command. The user can decide which components are not to be computed by interpolation by passing a list of indices to the parameter `BlackList`. A component of the solution is given the conventional value `"?"` meaning *non-computed* when either it cannot be computed because of too few interpolating points or the user desires not to compute it (it is in the `BlackList`).

An Example We can use `HybridNullSpace` to find the null space of matrices that are solved by the Mathematica built-in null space algorithm in a much longer time. For instance we can consider the system that has to be solved by Zeilberger's algorithm applied to the binomial coefficient to the fifth power, i. e. the problem of finding a special linear recurrence with polynomial coefficients in n and free of k for

$$\sum_{k=0}^{n} \binom{n}{k}^5. \tag{9}$$

Let us assume that we have the corresponding matrix, say, in `b5coefs`. Then we can solve the problem by using the command

```
HybridNullSpace[b5coefs,
      BlackList -> {2, 3, 4, 5, 6, 7, 8, 9, 10}]
```

Timings Here we give some timings in seconds of the built-in Mathematica command and `HybridNullSpace` for the linear systems involved in Zeilberger's fast algorithm for symbolic hypergeometric summation in the case of powers of binomials:

power	built-in	`HybridNullSpace`	`BlackList`
4	0.31	0.51	$\{2, \ldots, 8\}$
		5.04	$\{\}$
5	65.81	18.12	$\{2, \ldots, 10\}$
		30.95	$\{\}$
6	1721.76	1281.87	$\{2, \ldots, 15\}$
		1327.63	$\{\}$

These timings were obtained with Mathematica 4.0 under Linux on a Pentium II at 400Mhz.

Conclusion The method based on rational function interpolation performs much better than the method based on Cramer's rule and polynomial interpolation, in that it only reconstructs those components that require few homomorphic images.
The bottle-neck of the latter approach, as it is implemented now, is the interpolation step in which the extended Euclidean algorithm is invoked. Future versions of the package could use a more efficient interpolation algorithm and a parallelized interpolation of the components.

Acknowledgments

I thank my former advisor Prof. P. Paule at RISC-Linz and my advisor Prof. C. Traverso at the University of Pisa for their support.

References

[Car03] F. Caruso. *Polynomial Arithmetic and Linear Systems in Symbolic Summation*. PhD thesis, RISC-Linz (Austria), 2003. No. 03-01.

[GvzG99] J. Gerhard and J. von zur Gathen. *Modern Computer Algebra*. Cambridge University Press, 1999.

[Lip81] J.D. Lipson. *Algebra and Algebraic Computing*. Addison-Wesley, Reading, Massachusetts, 1981.

[PS95] P. Paule and M. Schorn. A Mathematica Version of Zeilberger's Algorithm for Proving Binomial Coefficient Identities. *J. of Symbolic Computation*, (20):673–698, 1995.

[PS00] C. Pau and W. Schreiner. Distributed Mathematica - User and Reference Manual. RISC Report 00-25, RISC, J.K. University, Linz, Austria, 2000.

[RZ] A. Riese and B. Zimmermann. Private communication.

[Zei90] D. Zeilberger. A fast algorithm for proving terminating hypergeometric identities. *Discrete Mathematics*, (80):207–211, 1990.