Solving Linear Systems of Equations Over Cyclotomic Fields

Liang Chen Michael Monagan

Computational Algebra Group
Centre for Experimental and Constructive Mathematics
Department of Mathematics
Simon Fraser University

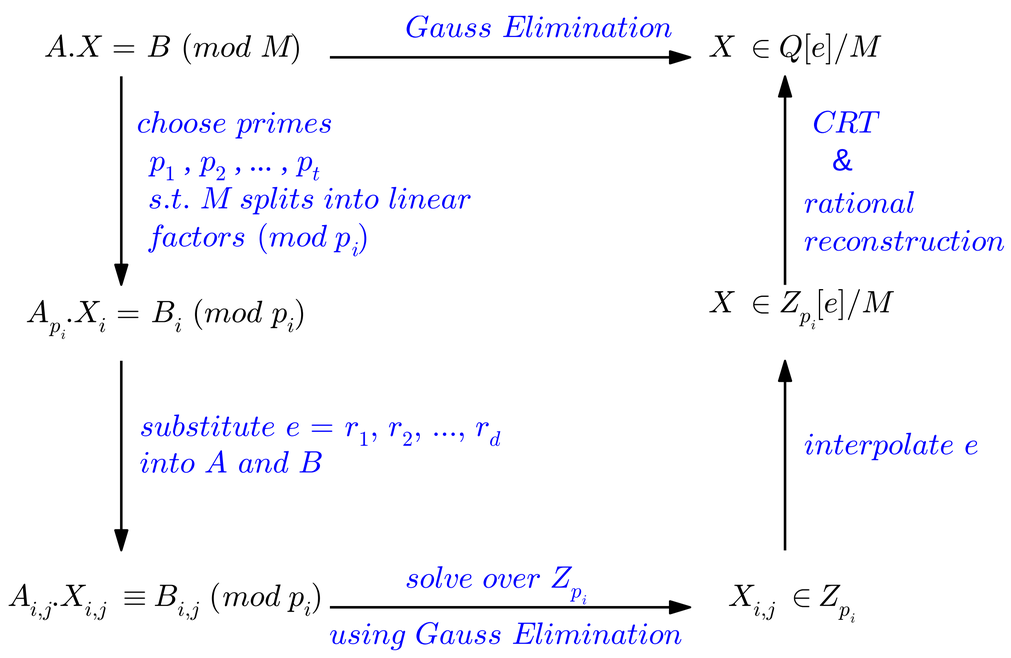
Abstract

Solving linear systems of equations over cyclotomic fields by directly applying Gauss Elimination is inefficient. We consider two approaches, namely, Chinese remaindering and p-adic lifting. Both of the approaches use rational reconstruction to recover the rational coefficients in solution vectors.

Notations

$\mathbf{e}$: a primitive m^{th} root of unity
$\mathbf{A}$: an n by n matrix over $\mathbb{Q}[e]$
$\mathbf{B}$: a vector of n elements over $\mathbb{Q}[e]$
$\mathbf{X}$: the solution vector which satisfies $\mathbf{A}.\mathbf{X} = \mathbf{B} \pmod{\mathbf{M}}$.
$\mathbf{M}$: minimal polynomial for e of degree d

Chinese Remaindering with Rational Reconstruction



Input: Matrix $\mathbf{A}$, Vector $\mathbf{B}$, Polynomial $\mathbf{M}$
Output: Vector $\mathbf{X}$ which satisfies $\mathbf{A}.\mathbf{X} = \mathbf{B}$ $(mod\ \mathbf{M})$

1. Clear fractions in Matrix A and Vector B
2. Generate suitable primes $p_1, p_2, \ldots, p_t$
3. **for** $i = 1$ **to** t **do**
   (a) find all roots $r_1, \ldots, r_d$ of M mod p_i
   (b) **for** $j = 1$ **to** d **do**
      i. substitute r_j into A and B
      ii. solve linear system $A_{i,j}.X_{i,j} \equiv B_{i,j}$ mod p_i for $X_{i,j}$
   (c) interpolate e from points r_j's and $X_{i,j}$'s
4. Apply Chinese Remainder Theorem to recover X mod $p_1 \times p_2 \times \cdots \times p_t$
5. Apply Rational Reconstruction algorithm to recover X over $\mathbb{Q}[e]/M$

Strength and Weakness

- We choose primes $p_1, p_2, ..., p_t$ to be half size of machine primes so that multiplication in $\mathbb{Z}_{p_i}$ is done in the hardware.
- The primes which can factor M into linear factors appear approximately one in every d primes.
- Since we do not know in advance how many primes is sufficient to recover the solution coefficients, we need to check the solution iteratively to see if we get the correct answer. We check the solution quadratically instead of linearly in the implementation to avoid doing too many unsuccessful rational reconstructions.

p-adic Lifting with Rational Reconstruction

Input: Matrix $\mathbf{A}$, Vector $\mathbf{B}$, Polynomial $\mathbf{M}$
Output: Vector $\mathbf{X}$ which satisfies $\mathbf{A}.\mathbf{X} = \mathbf{B} \pmod{\mathbf{M}}$

1. Clear fractions in Matrix A and Vector B
2. Find one suitable prime p
3. Find all roots $r_1, \ldots, r_d$ of M mod p
4. Set k := 1, $error_0 := B$
   (a) **Loop** until k reaches a certain bound
      i. **for** $i := 1$ **to** d **do**
         A. substitute r_i into A and $error_{k-1}$
         B. solve linear system $A_i.X_{k-1,i} = error_{k-1,i}$ over $\mathbb{Z}_p$
      ii. Interpolate e to obtain X_{k-1} from r_i's and $X_{k-1,i}$'s
      iii. Update $error_k$ and set $k := k + 1$
5. Obtain $X^{(k)} := X_0 + X_1 \times p + X_2 \times p^2 + \cdots + X_{k-1} \times p^{k-1}$
6. Apply rational reconstruction to recover X over $\mathbb{Q}[e]/M$

Strength and Weakness

Linear p-adic lifting: Same as the Chinese Remaindering approach, we require the prime to be half of the size of machine primes so that we can utilize the Modular package in Maple. By using p-adic lifting, we need only one such prime which splits M into linear factors over $\mathbb{Z}_p$. Since we do not know when to stop the lifting process, we need to check the solution periodically until either it returns the correct solution or "FAIL" if it reaches a bound. In this approach, we need to compute the error term which consumes the most of time.

Quadratic p-adic lifting: It is easy to lift the roots of M over $\mathbb{Z}_{p^i}$ to roots over $\mathbb{Z}_{p^{2i}}$. However, we can not use the Modular package since the calculation is done in $\mathbb{Z}_{p^i}$. In each iteration of quadratic lifting, it doubles the length of recovering coefficients which reduces time on computing the error term. The most time consuming part in this approach is to solve the linear systems over $\mathbb{Z}_{p^i}$ in each iteration.

Experiment

Let $M := e^6 + e^5 + e^4 + e^3 + e^2 + e + 1$
Let the entries in A and B be random polynomials in $\mathbb{Z}[e]/M$

Use Gauss Elimination:

matrix dimensions	Coefficient Length				
	2 digits	4 digits	8 digits	16 digits	32 digits
5	.039	.054	.094	.174	.401
10	.420	.713	1.395	3.33	9.445
20	6.517	13.63	35.04	106.2	362.6
40	171.4	460.3	1453	-	-
80	-	-	-	-	-

Use CRT & RR:

matrix dimensions	Coefficient Length				
	2 digits	4 digits	8 digits	16 digits	32 digits
5	.026	.062	.101	.182	.506
10	.068	.145	.325	.697	2.208
20	.310	.596	1.25	3.104	12.629
40	1.961	3.692	7.687	18.676	93.053
80	15.1	29.142	58.452	147.744	815.724

Use Linear p-adic Lifting & RR:

matrix dimensions	Coefficient Length				
	2 digits	4 digits	8 digits	16 digits	32 digits
5	.052	.070	.170	.425	1.086
10	.154	.271	.522	1.312	3.892
20	.538	.974	1.854	4.954	14.814
40	2.391	3.725	7.626	20.347	70.763
80	11.317	19.20	38.92	102.56	408.60

Data Analysis

Observing the trend of data, we learn that the CRT approach has advantage in computing small systems with large coefficients, and linear p-adic lifting approach has advantage in solving large systems with small coefficients. For example, let A be a 3×3 matrix, the coefficients be $10,000$ decimal digits long, and M be a cyclotomic polynomial of degree 2. It takes a modified CRT program 128 seconds to finish and a modified linear lifting program 4841 seconds to finish. However, the quadratic p-adic lifting program can do almost as good as the CRT one by taking 139 seconds to complete.

Maplesoft
command the brilliance
MITACS
NSERC CRSNG