Algoritmo para el cálculo de la matriz inversa de una matriz en *GF(p)*, ***p***-primo

Solution of a boundary problem Very Complex for hyperbolic equations: Case Zero Index

Pablo Freyre, Nelson Díaz (pfreyre@matcom.uh.cu)
Facultad de Matemática y Computación, Universidad de La Habana. Cuba

Resumen

En este trabajo se presenta un nuevo algoritmo que permite dada una matriz invertible, con sus elementos pertenecientes al campo de Galois *GF(p)*, *p*–primo, obtener su inversa. El algoritmo transforma la matriz escrita en su forma clásica, en polinomios y elementos del campo *GF(p)*, a partir de los cuales se calcula la inversa de la matriz. La ventaja de este algoritmo, con respecto a otros algoritmos, es que permite resolver el sistema de ecuaciones X = YA-1 con Y y A conocidos, donde A ∈ GLn(*GF(p)*)[1] y X, Y ∈ (*GF(p)*)n, sin necesidad de calcular explícitamente la matriz inversa, sino utilizando los polinomios asociados a la matriz A. El algoritmo está implementado en lenguaje Mathematica.

Palabras clave: Polinomios primitivos, sistemas de ecuaciones lineales, matriz inversa.

Abstract

In this work we present a new algorithm that allows, given an invertible matrix defined over *GF(p)*, p-prime, obtain their inverse. The algorithm transforms the matrix written in its classical form into polynomials and elements the field *GF(p)*, starting from which the inverse of the matrix is calculated. The advantage of this algorithm is that it allows to solve equations system in the form X = YA-1, with known Y and A, where A ∈ GLn (*GF(p)*) and Y, X ∈ (*GF(p)*)n, with no need to calculate explicitly the inverted matrix, but using instead the inverted polynomials associated with to matrix A. The algorithm is implemented on Mathematica language.

Key Words: Primitive polynomials, lineal equation systems, invertible matrix.

1. Introducción

En muchas aplicaciones prácticas se presenta la necesidad de resolver un sistema de ecuaciones lineales en un Campo de Galois *GF(p)*, *p*-primo. Reviste por tanto interés práctico el desarrollo de algoritmos eficientes para tales fines.

El objetivo de este trabajo es exponer la implementación, en lenguaje Mathematica, de un nuevo algoritmo que permite dado una matriz invertible, con sus elementos pertenecientes al campo *GF(p)*, obtener su inversa. El algoritmo resuelve el sistema de ecuaciones Y = XA cuando se conoce Y y A, donde A ∈ $GL_n(GF(p))$ y X , Y ∈ $(GF(p))^n$, sin necesidad de calcular explícitamente la matriz inversa.

En el trabajo de Freyre P., Díaz N. y Morgado E. R. (2009) se presenta este algoritmo en pseudocódigo con su fundamentación para el caso general en que las matrices se encuentran definidas en un campo finito arbitrario. En Freyre P. y Díaz N. (2007) se muestra este algoritmo implementado en lenguaje Mathematica para el caso particular en que las matrices están definidas en el campo *GF(*2*)*. Las operaciones que se deben

[1] $GL_n(GF(p))$ – Grupo Lineal General en *GF(p)*.

realizar en el algoritmo son el cálculo de la inversa de polinomios y la multiplicación de polinomios módulos polinomios primitivos para polinomios definidos en el campo *GF(p)*.

En el presente trabajo se muestran varios ejemplos de cálculo de la matriz inversa de una matriz invertible A y posteriormente dada la matriz A y el vector Y se resuelve el sistema de ecuaciones $X = YA^{-1}$. Los ejemplos se realizan para matrices definidas en *GF(p)* con $p = 2, 3, 5, y 7$.

2. Desarrollo

Los algoritmos que a continuación se citan, se encuentran implementados en lenguaje Mathematica y en ambos casos tenemos que:

n Es el grado de la matriz.

lpp Es la lista de polinomios primitivos a utilizarse en el algoritmo, que van en grado descendente desde *n* hasta 1. Los *n* polinomios primitivos seleccionados se calculan a *priori* y no hay restricción en su selección. (*ver* Lidl R. y Niederreiter H. (1994) y Peterson W.W. y Weldon J. E. (1972)).

3. Algoritmo para el cálculo de la matriz inversa

El algoritmo para el cálculo de la matriz inversa consta de dos pasos:

- Algoritmo que expresa la matriz a través de polinomios y elementos del campo *GF(p)*, *p*-primo.
- Algoritmo para el cálculo de la matriz inversa.

3.1 Algoritmo que expresa la matriz a través de polinomios y elementos de *GF(p)*

Programación del algoritmo:

m Es la matriz a la que se le va a calcular su inversa.

vbc Son los valores de los elementos del campo *GF(p)* y de los coeficientes de los polinomios definidos en el campo *GF(p)* asociados a la matriz *m*.

```
Clear[Creavbc]
Creavbc[n _ , i _ , v _ , vbc _ , lpp _ ] :=
Block[{x = 0, t, z, y = PadLeft[{}, n]},
  z = lpp[[i]][Take[vbc[[i]], {i, n}]];
  t = lpp[[i]][Take[y, {i, n}]];
  If[TrueQ[z[[1]] != t[[1]]], t = lpp[[i]][Take[v,
  {i, n}]]*(z^-1);
    If[TrueQ[t == 0], t = lpp[[i]][Take[y, {i,
    n}]]];
      x = lpp[[1]][Take[v, {1, i - 1}]] -
        lpp[[1]][Take[vbc[[i]], {1, i -
        1}]]*lpp[[1]][t[[1]][[{1}]]];
    ];
    If[TrueQ[x == 0], x = lpp[[1]][PadLeft[{},
    n]]];
    Return[Join[Take[x[[1]], {1, i - 1}],
    t[[1]]]];
]

Clear[Fpolinomial]
Fpolinomial[n _ , m _ , lpp _ ] :=
Block[{vbc = {}, i, j, vec, y = PadLeft[{}, n]},
  For[j = 1, j <= n, j++,
    i = 0; vec = m[[j]];
    While[(i = i + 1) < j,
      vec = Creavbc[n, i, vec, vbc, lpp];
      If[Take[vec, {i, n}] == Take[y, {i, n}],
      Print["Verifique si la matriz es
            inversible."]; Return[y]]
    ];
    If[Take[vec, {i, n}] == Take[y, {i, n}],
    Print["Verifique si la matriz es inversi
          ble."]; Return[y]];
    AppendTo[vbc, vec]
  ];
  Return[vbc];
]
```

4. Algoritmo para el cálculo de la matriz inversa

Programación del algoritmo:

vbc Son los valores de los elementos del campo *GF(p)* y de los coeficientes de los polinomios definidos en el campo *GF(p)* asociados a la matriz anterior.

m Es la matriz inversa.

```
Clear[ILb]
ILb[n _ , i _ , v _ , vbc _ , lpp _ ] :=
```

```
Block[{x, t, z},
  t = lpp[[i]][Take[v, {i, n}]]*(lpp[[i]][Take[-
  vbc[[i]], {i, n}]]^-1);
  If[TrueQ[t == 0], t = lpp[[i]][PadLeft[{}, n
  + 1 - i]]];
  x = lpp[[1]][Take[v, {1, i - 1}]] -
      lpp[[1]][Take[vbc[[i]], {1, i -
  1}]]*lpp[[1]][t[[1]][[{1}]]];
  If[TrueQ[x == 0], x = lpp[[1]][PadLeft[{},
  n]]];
  Return[Join[Take[x[[1]], {1, i - 1}], t[[1]]]];
]

Clear[Invmatriz]
Invmatriz[n _ , vbc _ , lpp _ ] :=
Block[{m = {}, v = IdentityMatrix[n], i, j,
vec},
  For[j = 1, j <= n, j++,
    i = 0; vec = v[[j]];
    While[(i = i + 1) < n + 1,vec = ILb[n, i,
    vec, vbc, lpp]];
    AppendTo[m, vec]
  ];
  Return[m];
]
```

5. Algoritmo para resolver la ecuacion $X = YA^{-1}$ conociendo los valores de la matriz a y el vector Y

Programación del algoritmo:

y Valor del vector Y.

m Es la matriz A.

x Valor del vector X.

```
Clear[Resolver]
Resolver[n _ ,m _ ,y _ ,lpp _ ]:=
Block[{vbc, i, x},
  vbc=Fpolinomial[n,m,lpp];
  If[TrueQ[vbc    PadLeft[{},n]],Prin-
  t[«Verifique si la matriz
  inversible.”];Return[vbc]];
  i=0;x=y;While[(i=i+1)<n+1,x=ILb[n,i,x,vbc,l-
  pp]];
  Return[x];
]
```

En los algoritmos anteriores se han utilizado las funciones propias del lenguaje Mathematica para el cálculo de la inversa de polinomios y la multiplicación de polinomios módulos polinomios primitivos, pero en una implementación específica se puede aumentar la velocidad de procesamiento utilizando los algoritmos que se reportan en Menezes, Van Oorschot y Varston (1996) para la multiplicación y cálculo de la inversa de polinomios módulo polinomios primitivos definidos en el campo *GF(p)*, *p*-primo, con polinomios primitivos con un número mínimo de coeficientes.

Ejemplos:

En los ejemplos que se exponen a continuación primero se calcula la matriz inversa de la matriz m y posteriormente dada la matriz m y el vector *Y* se resuelve el sistema de ecuaciones $X = Ym^{-1}$ sin necesidad de calcular explícitamente la matriz inversa. Los polinomios primitivos utilizados en los ejemplos fueron tomados de la tabla C.2 de Peterson W. W. y Weldon J. E. (1972) para $p = 2$ y de la tabla F de Lidl R. y Niederreiter H. (1994) para $p = 3$, 5 y 7.

Ejemplos:

1. Cálculo de la matriz inversa de la matriz m definida en *GF(2)* que se expresa a continuación:

$$\begin{pmatrix} 0&1&1&0&0&1&1&0&0&1\\ 0&1&1&1&0&1&1&0&1&1\\ 0&1&1&0&1&0&0&1&0&0\\ 0&0&1&1&0&1&1&1&1&1\\ 0&1&0&1&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&0&1&1\\ 0&1&0&0&1&0&1&0&0&0\\ 1&0&0&0&1&1&0&0&0&1\\ 0&0&1&1&1&0&1&0&0&0\\ 1&1&0&1&0&0&0&0&0&1 \end{pmatrix}$$

Los polinomios primitivos son: $1 + x^3 + x^{10}$; $1 + x^4 + x^9$; $1 + x^2 + x^3 + x^4 + x^8$; $1 + x^3 + x^7$; $1 + x + x^6$; $1 + x^2 + x^5$; $1 + x + x^4$; $1 + x + x^3$; $1 + x + x^2$; $1 + x$.

```
<<Algebra`FiniteFields`
lpp = {GF[2,{1,0,0,1,0,0,0,0,0,0,1}],GF[2,{1,0,0,0,1,0,0,0,
            0,1}],GF[2,{1,0,1,1,1,0,0,0, 1}],GF[2,{1,0,0,
            1,0,0,0,1}],GF[2,{1,1,0,0,0,0,1}],GF[2,{1,0,1,
            0,0,1}],GF[2,{1,1,0,0,1}],GF[2,{1,1,0,1}],
       GF[2,{1,1,1}],GF[2,{1,1}]}
m   = {{0,1,1,0,0,1,1,0,0,1},{0,1,1,1,0,1,1,0,1,1},{0,1,1,0,
        1,0,0,1,0,0},{0,0,1,1,0,1,1,1,1,1},{0,1,0,1,0,0,0,0,
      0,1},{0,0,0,0,0,0,0,0,1,1},{0,1,0,0,1,0,1,0,0,0},{1,
      0,0,0,1,1,0,0,0,1},{0,0,1,1,1,0,1,0,0,0},{1,1,0,1,0,
      0,0,0,0,1}}
vbc = Fpolinomial[10,m,lpp];
t = Invmatriz[10,vbc,lpp];
MatrixForm[%]
```

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1
\end{pmatrix}$$

Resolución del sistema $x = y\ m^{-1}$ utilizando el algoritmo # 2 y teniendo como datos de entrada: la matriz m expuesta al comienzo del ejemplo y el vector y:

```
y = {1,0,1,1,1,0,0,1,1,1}
x = Resolver[10, m, y, lpp]
x = {1,0,1,0,0,1,0,1,1,0}
```

2. Cálculo de la matriz inversa de la matriz m definida en GF(2) que se expresa a continuación

$$\begin{pmatrix}
0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 1
\end{pmatrix}$$

Los polinomios primitivos son: $1 + x^2 + x^5$; $1 + x + x^4$; $1 + x + x^3$; $1 + x + x^2$; 1+x.

```
<<Algebra`FiniteFields`
lpp = {GF[2,{1,0,1,0,0,1}],GF[2,{1,1,0,0,1}],GF[2,{1,1,
            0,1}],GF[2,{1,1,1}],GF[2,{1,1}]}
m = {{0,0,1,0,0},{0,1,0,1,0},{1,1,0,0,0},{0,0,1,1,1},{1,0,1,1,1}}
vbc = Fpolinomial[5,m,lpp];
t = Invmatriz[5,vbc,lpp];
MatrixForm[%]
```

$$\begin{pmatrix}
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 1
\end{pmatrix}$$

Resolución del sistema $x = y\ m^{-1}$ utilizando el algoritmo # 2 y teniendo como datos de entrada: la matriz m expuesta al comienzo del ejemplo y el vector y:

```
y = {0,1,1,0,1}
x = Resolver[5, m, y, lpp]
x = {0,1,0,1,0}
```

3. Cálculo de la matriz inversa de la matriz m definida en GF(3) que se expresa a continuación

$$\begin{pmatrix}
0 & 2 & 2 & 2 & 0 & 1 & 2 & 2 & 0 & 2 & 2 \\
0 & 2 & 1 & 1 & 1 & 1 & 2 & 2 & 0 & 1 & 0 \\
0 & 1 & 0 & 2 & 0 & 2 & 1 & 1 & 1 & 2 & 0 \\
0 & 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
2 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 2 \\
1 & 1 & 2 & 2 & 2 & 1 & 2 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 2 & 1 & 0 & 0 & 0 \\
1 & 2 & 2 & 1 & 2 & 0 & 2 & 2 & 1 & 0 & 2 \\
0 & 1 & 0 & 0 & 2 & 1 & 0 & 2 & 0 & 0 & 1 \\
2 & 0 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 1 \\
0 & 2 & 0 & 1 & 1 & 2 & 2 & 0 & 0 & 0 & 0
\end{pmatrix}$$

Los polinomios primitivos son:

$1 + x^4 + x^{10} + x^{11}$; $2 + x^7 + x^9 + x^{10}$;

$1 + x^5 + x^7 + x^9$; $2 + x^5 + x^8$; $1 + x^4 + x^6 + x^7$; $2 + x^5 + x^6$;

$1 + x^2 + x^4 + x^5$; $2 + x^3 + x^4$; $1 + 2x^2 + x^3$; $2 + x + x^2$; 1+x.

```
<<Algebra`FiniteFields`
lpp = {GF[3,{1,0,0,0,1,0,0,0,0,0,1,1}],GF[3,{2,0,0,0,0,0,0,
            1,0,1,1}],GF[3,{1,0,0,0,0,1,0,1,0,1}],GF[3,{2,
            0,0,0,0,1,0,0,1}],GF[3,{1,0,0,0,1,0,1,1}],
       GF[3,{2,0,0,0,0,1,1}],GF[3,{1,0,1,0,1,1}],GF[3,{2,0,
       0,1,1}],GF[3,{1,0,2,1}],GF[3,{2,1,1}],GF[3,{1,1}]}
m = {{0,2,2,2,0,1,2,2,0,2,2},{0,2,1,1,1,1,2,2,0,1,0},{0,1,
      0,2,0,2,1,1,1,2,0},{0,2,0,0,0,0,1,0,0,1,1},{2,0,2,0,
      0,2,0,0,0,2,2},{1,1,2,2,2,1,2,1,0,1,0},{1,1,0,0,1,0,
      2,1,0,0,0},{1,2,2,1,2,0,2,2,1,0,2},{0,1,0,0,2,1,0,2,
      0,0,1},{2,0,0,1,2,0,1,2,0,1,1},{0,2,0,1,1,2,2,0,0,0,
      0}}
vbc = Fpolinomial[11,m,lpp];
t = Invmatriz[11,vbc,lpp];
MatrixForm[%]
```

$$\begin{pmatrix}
2 & 1 & 2 & 2 & 0 & 1 & 2 & 1 & 1 & 0 & 0 \\
0 & 2 & 1 & 0 & 1 & 2 & 1 & 2 & 2 & 1 & 1 \\
2 & 2 & 1 & 2 & 2 & 1 & 2 & 2 & 1 & 0 & 0 \\
1 & 2 & 2 & 2 & 2 & 1 & 0 & 1 & 2 & 0 & 2 \\
0 & 1 & 0 & 2 & 0 & 1 & 1 & 0 & 2 & 2 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 2 & 0 & 2 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 2 & 1 & 2 & 0 & 1 & 1 \\
2 & 2 & 1 & 0 & 0 & 1 & 1 & 2 & 1 & 1 & 2 \\
1 & 1 & 0 & 2 & 1 & 1 & 0 & 1 & 0 & 1 & 2 \\
0 & 2 & 0 & 1 & 1 & 1 & 1 & 0 & 2 & 1 & 1 \\
2 & 0 & 0 & 0 & 2 & 2 & 2 & 0 & 0 & 2 & 2
\end{pmatrix}$$

Resolución del sistema $x = y\ m^{-1}$ utilizando el algoritmo # 2, con datos de entrada: la matriz m del comienzo del ejemplo y el vector y:

```
y = {2,1,2,1,2,0,0,1,0,0,0}
x = Resolver[11, m, y, lpp]
x = {2,2,1,1,1,1,0,2,1,0,1}
```

4. Cálculo de la matriz inversa de la matriz ***m*** definida en *GF(5)* que se expresa a continuación

$$\begin{pmatrix} 3 & 1 & 0 & 4 & 4 & 0 & 4 & 0 & 2 \\ 2 & 3 & 0 & 2 & 4 & 2 & 2 & 0 & 0 \\ 4 & 1 & 4 & 2 & 0 & 0 & 3 & 0 & 4 \\ 3 & 0 & 0 & 3 & 1 & 1 & 4 & 4 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 3 & 3 \\ 4 & 2 & 2 & 2 & 3 & 3 & 0 & 0 & 1 \\ 3 & 0 & 1 & 4 & 4 & 2 & 1 & 3 & 4 \\ 4 & 3 & 4 & 0 & 0 & 3 & 3 & 1 & 2 \\ 0 & 1 & 3 & 2 & 0 & 1 & 4 & 2 & 3 \end{pmatrix}$$

Los polinomios primitivos son:

$3 + x^6 + x^7 + x^9$; $3 + x^3 + x^5 + x^8$; $2 + x^6 + x^7$; $2 + x^5 + x^6$;

$2 + x^2 + x^5$; $3 + x + x^3 + x^4$; $2 + x^2 + x^3$; $2 + x + x^2$; 2+x.

```
<<Algebra`FiniteFields`
lpp = {GF[5,{3,0,0,0,0,0,1,1,0,1}],GF[5,{3,0,0,1,0,1,0,0,
            1}],GF[5,{2,0,0,0,0,0,1,1}],GF[5,{2,0,0,0,0,1,
            1}],GF[5,{2,0,1,0,0,1}],GF[5,{3,1,0,1,1}],
        GF[5,{2,0,1,1}],GF[5,{2,1,1}],GF[5,{2,1}]}
m = {{3,1,0,4,4,0,4,0,2},{2,3,0,2,4,2,2,0,0},{4,1,4,2,0,0,
      3,0,4},{3,0,0,3,1,1,4,4,0},{1,1,0,1,1,1,0,3,3},{4,2,
      2,2,3,3,0,0,1},{3,0,1,4,4,2,1,3,4},{4,3,4,0,0,3,3,1,
      2},{0,1,3,2,0,1,4,2,3}}
vbc = Fpolinomial[9,m,lpp];
t = Invmatriz[9,vbc,lpp];
MatrixForm[%]
```

$$\begin{pmatrix} 2 & 4 & 2 & 2 & 4 & 3 & 4 & 0 & 4 \\ 0 & 1 & 1 & 2 & 0 & 1 & 4 & 3 & 1 \\ 4 & 4 & 4 & 1 & 3 & 2 & 2 & 1 & 0 \\ 0 & 4 & 3 & 0 & 3 & 2 & 0 & 1 & 0 \\ 3 & 1 & 4 & 1 & 2 & 3 & 2 & 0 & 2 \\ 3 & 1 & 0 & 4 & 4 & 4 & 1 & 1 & 4 \\ 4 & 3 & 3 & 4 & 0 & 1 & 0 & 1 & 4 \\ 3 & 3 & 1 & 1 & 2 & 1 & 0 & 3 & 4 \\ 1 & 0 & 4 & 1 & 4 & 3 & 3 & 2 & 4 \end{pmatrix}$$

Resolución del sistema $x = y\ m^{-1}$ utilizando el algoritmo # 2 y teniendo como datos de entrada: la matriz m expuesta al comienzo del ejemplo y el vector y:

```
y = {3,2,1,2,0,4,1,1,3}
x = Resolver[11, m, y, lpp]
x = {2,1,4,0,1,4,0,3,0}
```

5. Cálculo de la matriz inversa de la matriz ***m*** definida en *GF(7)* que se expresa a continuación

$$\begin{pmatrix} 6 & 1 & 3 & 3 & 4 & 2 & 3 & 0 & 0 & 3 \\ 2 & 2 & 3 & 1 & 5 & 1 & 3 & 1 & 6 & 6 \\ 2 & 2 & 4 & 2 & 5 & 5 & 1 & 6 & 4 & 0 \\ 2 & 1 & 1 & 5 & 1 & 6 & 2 & 3 & 4 & 6 \\ 2 & 1 & 5 & 6 & 0 & 0 & 6 & 1 & 4 & 1 \\ 2 & 1 & 6 & 0 & 3 & 5 & 6 & 6 & 1 & 6 \\ 0 & 4 & 3 & 2 & 2 & 6 & 6 & 3 & 1 & 2 \\ 3 & 6 & 4 & 3 & 2 & 0 & 1 & 4 & 5 & 1 \\ 1 & 6 & 2 & 3 & 5 & 6 & 6 & 4 & 5 & 5 \\ 2 & 4 & 4 & 2 & 0 & 0 & 2 & 3 & 4 & 4 \end{pmatrix}$$

Los polinomios primitivos son:

$3 + x^8 + x^9 + x^{10}$; $2 + x^3 + x^8 + x^9$; $3 + x^7 + x^8$; $4 + x^5 + x^7$;

$3 + x^4 + x^5 + x^6$; $4 + x^4 + x^5$; $3 + x^2 + x^3 + x^4$; $2 + x + x^2 + x^3$;

$3 + x + x^2$; 2+x.

```
<<Algebra`FiniteFields`
lpp = {GF[7,{3,0,0,0,0,0,0,1,1,1}],GF[7,{2,0,0,1,0,0,0,0,
            1,1}],GF[7,{3,0,0,0,0,0,0,1,1}],GF[7,{4,0,0,0,
            0,1,0,1}],GF[7,{3,0,0,0,1,1,1}],GF[7,{4,0,0,0,
            1,1}],GF[7,{3,0,1,1,1}],GF[7,{2,1,1,1}],GF[7,{
            3,1,1}],GF[7,{2,1}]}
m = {{6,1,3,3,4,2,3,0,0,3},{2,2,3,1,5,1,3,1,6,6},{2,2,4,2,
      5,5,1,6,4,0},{2,1,1,5,1,6,2,3,4,6},{2,1,5,6,0,0,6,1,
      4,1},{2,4,6,0,3,5,6,6,1,6},{0,4,3,2,2,6,6,3,1,2},{3,
      6,4,3,2,0,1,4,5,1},{1,6,2,3,5,6,6,4,5,5},{2,4,4,2,0,
      0,2,3,4,4}}
vbc = Fpolinomial[10,m,lpp];
t = Invmatriz[10,vbc,lpp];
MatrixForm[%]
```

$$\begin{pmatrix} 4 & 4 & 1 & 4 & 3 & 0 & 2 & 4 & 4 & 0 \\ 1 & 5 & 2 & 1 & 1 & 4 & 6 & 5 & 2 & 0 \\ 0 & 3 & 3 & 4 & 1 & 1 & 2 & 0 & 3 & 4 \\ 5 & 2 & 3 & 6 & 5 & 4 & 0 & 6 & 6 & 1 \\ 4 & 2 & 1 & 0 & 2 & 4 & 3 & 1 & 4 & 0 \\ 5 & 0 & 5 & 4 & 5 & 0 & 3 & 3 & 6 & 2 \\ 6 & 5 & 5 & 6 & 5 & 4 & 2 & 6 & 5 & 5 \\ 6 & 3 & 0 & 2 & 1 & 6 & 6 & 5 & 2 & 0 \\ 3 & 6 & 3 & 2 & 0 & 1 & 5 & 3 & 4 & 4 \\ 5 & 1 & 5 & 1 & 6 & 3 & 5 & 3 & 3 & 5 \end{pmatrix}$$

Resolución del sistema $x = y\ m^{-1}$ utilizando el algoritmo # 2 y teniendo como datos de entrada: la matriz m expuesta al comienzo del ejemplo y el vector y:

```
y = {5,6,5,6,5,2,2,4,3,1}
x = Resolver[10, m, y, lpp]
x = {3,2,5,5,4,6,6,1,1,1}
```

Bibliografía

FREYRE P., DÍAZ N Y MORGADO E. R. (2009). "Some algorithms related to matrices with entries in finite field". Journal of Discrete Mathematical Science and Cryptography. India.

FREYRE P. Y DÍAZ N. (2007). "Nuevo Algoritmo para el Cálculo de la Matriz Inversa". Revista Investigación Operacional. Vol. 28. No. 2. 179-185. Habana.

GOLOMB W. S. (1982). "Shift Register Secuences". Aegean Park Press. California.

MENEZES A., VAN OORSCHOT P. AND VARSTONE S. (1996). "Handbook of Applied Cryptography". CRC. Press.

LIDL R. Y NIEDERREITER H. (1994). "Introduction to Finite Fields and their Applications". Cambridge University.

PETERSON W.W. Y WELDON J. E. (1972). "Error-Correcting Codes". John Wiley and Sons, Inc. New York. 2ed.