4 Las formas irreducible y de Jordan

La tercera parte del paquete está destinada a calcular la forma de Jordan de una matriz sobre el cuerpo de escisión apropiado y otras formas canónicas, que exigen factorización del polinomio característico. Cubre los tópicos del capítulo II, párrafos 5-6 del libro

Procedimientos simbólicos en ALGEBRA LINEAL

El procedimiento JORDAN de este archivo está ya en el paquete “linalg” de MAPLE para el caso de característica cero; este archivo incluye característica arbitraria, más aún el algoritmo implementado es diferente y la matriz P , de cambio de base, obtenida es más sencilla que la que MAPLE da por defecto.

4.1 Principales procedimientos

REDUCE **Calcula una matriz semejante a una compañera de polinomios mínimos primos entre sí**

Diferentes maneras de utilización:

- REDUCE(C)
- REDUCE(C,K)
- REDUCE(C,P)
- REDUCE(C,P,K)
- REDUCE(C,inter)

Argumentos:

- C - una matriz compañera.
- (opcional) P - un símbolo (preferiblemente no “ligado”).
- (opcional) K - un número primo o un conjunto formado por números algebraicos, o un número primo y un conjunto formado por elementos algebraicos sobre Z/pZ .
- (opcional) inter - exactamente la expresión inter.

Observación importante: Puede ser muy recomendable "entrecomillar" la expresión `inter`, mediante la inserción de apóstrofes.

Descripción:

La función **REDUCE** implementa los procedimientos de las páginas 114 y 115 del libro, para calcular una matriz semejante a una compañera, que es suma diagonal de matrices compañeras con polinomios mínimos primos entre sí.

- **REDUCE(C)** factoriza sobre $\mathbf{Q}$, el polinomio característico de C y calcula las matrices compañeras asociadas a estos factores.
- **REDUCE(C,P)** factoriza sobre $\mathbf{Q}$, el polinomio característico de A , calcula las matrices compañeras asociadas a estos factores dando la matriz bloc-digonal C correspondiente, y aloja en la variable P una matriz P regular tal que $P^{-1}CP = C^*$. Para obtener la matriz P , desde el *prompt* de Maple solicítese `op(P)`.
- **REDUCE(C,K)** factoriza sobre el cuerpo K .
- **REDUCE(C,P,K)** factoriza sobre el cuerpo K y aloja en la variable P una matriz P regular de cambio de base. Para obtener la matriz P , desde el *prompt* de Maple solicítese `op(P)`.
- **REDUCE(C,inter)** ejecuta el programa de modo interactivo a efectos de aprendizaje. Esta modalidad permitirá al usuario elegir la factorización coprima que desee.

Ejemplos:

```
> C:=companion(x^4-16,x); > M:=REDUCE(C,P);
```

$$M := \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & -4 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

> op(P);

$$\begin{bmatrix} 8 & -8 & -4 & 0 \\ 4 & 4 & 0 & -4 \\ 2 & -2 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

> equal(evalm(inverse(P)&*C&*P),M);
true

> N:=REDUCE(C,Q,{1});

$$N := \begin{bmatrix} -2 & 0 & 0 & 0 \\ 0 & 2\sqrt{-1} & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & -2\sqrt{-1} \end{bmatrix}$$

> op(Q);

$$\begin{bmatrix} -8 & -8\sqrt{-1} & 8 & 8\sqrt{-1} \\ 4 & -4 & 4 & -4 \\ -2 & 2\sqrt{-1} & 2 & -2\sqrt{-1} \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

> equal(evalm(inverse(Q)&*C&*Q),N);
true

> J:=REDUCE(C,R,5);

$$J := \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> op(R);
```

$$\begin{bmatrix} 2 & 4 & 3 & 1 \\ 4 & 1 & 4 & 1 \\ 3 & 4 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

```
> H:=evalm(inverse(R)*C*R):
```

```
> equal(H,J);
```

```
false
```

```
> L:=modulariza(H,5):
```

```
> equal(L,J);
```

```
true
```

```
> REDUCE(C,inter); # El programa comenzará a ejecutar una sesión inter-
activa ante el usuario, solicitando a éste datos y opciones relativas al trabajo a
realizar. Permite elegir la factorización coprime deseada.
```

JORDAN Calcula la forma de Jordan de una matriz

Diferentes maneras de utilización:

- JORDAN(A)
- JORDAN(A,p)
- JORDAN(A,P)
- JORDAN(A,P,p)
- JORDAN(A,inter)

Argumentos:

- A - una matriz cuadrada.
- (opcional) P - un símbolo (preferiblemente no “ligado”).
- (opcional) p - un número primo
- (opcional) inter - exactamente la expresión inter.

Descripción:

La función **JORDAN** implementa los procedimientos de la página 123 del libro, para calcular la forma de Jordan de una matriz

En el caso de característica cero, por razones de eficiencia, se llama al procedimiento `splits` de MAPLE para calcular un cuerpo de escisión del polinomio característico de A .

Para el caso de característica positiva se ha incluido un procedimiento para calcular el cuerpo de escisión sobre $\mathbf{Z}_p$ del polinomio característico; debido a que se usa la función `Factors` de MAPLE es necesario que sus factores irreducibles contengan en sus coeficientes a lo sumo un elemento algebraico sobre $\mathbf{Z}_p$.

- `JORDAN(A)` calcula, en la extensión apropiada de $\mathbf{Q}$, la forma¹ de Jordan de A .
- `JORDAN(A,p)` calcula, en la extensión apropiada de $\mathbf{Z}_p$, la forma de Jordan de A .
- `JORDAN(A,P)` calcula, en la extensión apropiada de $\mathbf{Q}$, la forma de Jordan J de la matriz entrada A , y aloja en la variable P una matriz P regular tal que $P^{-1}AP = J$. Para obtener la matriz P , desde el *prompt* de Maple solicítese `op(P)`.
- `JORDAN(A,P,p)` calcula, en la extensión apropiada de $\mathbf{Z}_p$, la forma de Jordan J de la matriz entrada A , y aloja en la variable P una matriz P regular tal que $P^{-1}AP = J$. Para obtener la matriz P , desde el *prompt* de Maple solicítese `op(P)`.
- Si se incluye el parámetro opcional `inter`, se ejecuta el programa de modo interactivo a efectos de aprendizaje.

Ejemplos:

```
> A:=companion((x**2-2)*(x**2-3),x);
```

$$A := \begin{bmatrix} 0 & 0 & 0 & -6 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

```
> J:=JORDAN(A,P);
```

¹salvo permutación de los bloques de Jordan

$$J = \begin{bmatrix} -\sqrt{3} & 0 & 0 & 0 \\ 0 & \sqrt{2} & 0 & 0 \\ 0 & 0 & -\sqrt{2} & 0 \\ 0 & 0 & 0 & \sqrt{3} \end{bmatrix}$$

> op(P);

$$\begin{bmatrix} 2\sqrt{3} & -3\sqrt{2} & 3\sqrt{2} & -2\sqrt{3} \\ -2 & -3 & -3 & -2 \\ -\sqrt{3} & \sqrt{2} & -\sqrt{2} & \sqrt{3} \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

> H:= evalm(inverse(P)&*A&*P):

> equal(simplify(H),J);

true

> JORDAN(A,3);

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & \alpha_{11} & 0 \\ 0 & 0 & 0 & 2\alpha_{11} \end{bmatrix}$$

> op(alpha11);

$$-Z^2 + 1$$

> J:=JORDAN(A,P,3):

> op(P);

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & \alpha_{11} & 2\alpha_{11} \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

> M:=evalm(inverse(P)&*A&*P):N:=map(simplify,M):Q:=modulariza(N,3):

> equal(Q,J);

true

> J:=JORDAN(A,inter): El programa comenzará a ejecutar una sesión interactiva ante el usuario, solicitando a éste datos y opciones relativas al trabajo a realizar.

> B:=companion((x-a/b)*(x-c/d)*(x-3*e/f),x):

> J:=JORDAN(B,P);

$$J := \begin{bmatrix} 3e/f & 0 & 0 \\ 0 & a/b & 0 \\ 0 & 0 & c/d \end{bmatrix}$$

> op(P);

$$\begin{bmatrix} \frac{ac}{bd} & \frac{3ce}{df} & \frac{3ea}{fb} \\ -\frac{ad+cb}{bd} & -\frac{cf+3ed}{df} & -\frac{3eb+fa}{fb} \\ 1 & 1 & 1 \end{bmatrix}$$

> H:=evalm(inverse(P)&*B&*P):

> M:=map(simplify,H):

> equal(M,J);

true

Compare el lector estos cálculos con los de la primitiva de MAPLE:

> linalg[jordan](B,Q):

> op(Q);

$$\begin{bmatrix} -3 \frac{aed^2}{acdf-3aed^2-c^2bf+3cbde} & 3 \frac{ceb^2}{-fcab+fa^2d+3ceb^2-3deab} & \frac{acf^2}{acf^2-3efad-3efcb+9e^2bd} \\ \frac{d^2(3eb+fa)}{acdf-3aed^2-c^2bf+3cbde} & \frac{b^2(cf+3ed)}{-fcab+fa^2d+3ceb^2-3deab} & \frac{f^2(ad+cb)}{acf^2-3efad-3efcb+9e^2bd} \\ -\frac{d^2fb}{acdf-3aed^2-c^2bf+3cbde} & \frac{b^2df}{-fcab+fa^2d+3ceb^2-3deab} & \frac{dbf^2}{acf^2-3efad-3efcb+9e^2bd} \end{bmatrix}$$

> J:=JORDAN(B,P,3);

$$J := \begin{bmatrix} a/b & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & c/d \end{bmatrix}$$

> op(P);

$$\begin{bmatrix} 0 & \frac{ac}{bd} & 0 \\ 2\frac{c}{d} & \frac{2ad+2cb}{bd} & 2\frac{a}{b} \\ 1 & 1 & 1 \end{bmatrix}$$

El usuario que desee comprobar los cálculos no olvide que estamos trabajando en característica 3.

Jordan Calcula la forma de Jordan de una matriz

Se trata de la variante de la función JORDAN, en la que los bloques de Jordan aparecen traspuestos, es decir con los 1 encima de la diagonal. Esta función puede ser preferida por algunos usuarios.

Ejemplo:

> A:=companion((x**2-2)*(x**2-3),x):

> J:=Jordan(A,P,3);

$$J := \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_{11} & 0 \\ 0 & 0 & 0 & 2\alpha_{11} \end{bmatrix}$$

> op(P);

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & \alpha_{11} & 2\alpha_{11} \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

IRREDUCIBLE **Calcula la forma irreducible de una matriz**

Diferentes maneras de utilización:

- IRREDUCIBLE(A)
- IRREDUCIBLE(A,K)
- IRREDUCIBLE(A,P)
- IRREDUCIBLE(A,P,K)
- IRREDUCIBLE(A,inter)

Argumentos:

- A - una matriz cuadrada.
- (opcional) P - un símbolo (preferiblemente no “ligado”).
- (opcional) K - un número primo o un conjunto de elementos algebraicos² sobre $\mathbf{Q}$ o un conjunto formado por un número primo y elementos algebraicos² sobre el cuerpo primo.
- (opcional) inter - exactamente la expresión inter.

Descripción:

La función **IRREDUCIBLE** implementa los procedimientos de la página 116 del libro, calculando una matriz $G = \text{diag}[C(p_1), \dots, C(p_r)]$ semejante a la matriz entrada A, y alojando en la variable P una matriz P regular tal que $P^{-1}AP = G$, donde $C(p_i)$ es la matriz compañera del polinomio p_i y $(p_1, \dots, p_r)$ es la familia de divisores elementales de A. Para obtener la matriz P, desde el *prompt* de Maple solicítese op(P).

Sea $A = (a_{ij})$.

- IRREDUCIBLE(A) e IRREDUCIBLE(A,P) efectúan los cálculos en la extensión $\mathbf{Q}(a_{ij})$.
- Si $K = \{\alpha_1, \dots, \alpha_n\}$ es un conjunto de números algebraicos sobre $\mathbf{Q}$, los procedimientos IRREDUCIBLE(A,K) e IRREDUCIBLE(A,P,K) efectúan los cálculos en en la extensión $\mathbf{Q}(a_{ij}, \alpha_1, \dots, \alpha_n)$

²que MAPLE acepte como tales

- Si K contiene un elemento algebraico sobre $\mathbf{Z}_p$, `IRREDUCIBLE(A,P,K)` efectúa los cálculos en la extensión $\mathbf{Z}_p(a_{ij})$; debido a que se usa la función `Factors` de `MAPLE`, los términos a_{ij} de la matriz A pueden incluir a lo sumo un número algebraico en forma de `RootOf`. Para el caso $K = \{p, \alpha\}$, el ordenador puede tomarse su tiempo y memoria. Teste el usuario la matriz

$$\begin{bmatrix} a/b & c/d \\ e/f & g/h \end{bmatrix} \text{ sobre } \mathbf{Z}_5(\alpha) \quad \alpha^2 - 2 = 0$$

- Si se incluye el parámetro opcional `inter`, se ejecuta el programa de modo interactivo a efectos de aprendizaje.

Ejemplos:

```
> B:=companion((x**2-2)*(x**2-3),x):
```

```
> G:=IRREDUCIBLE(B,P,{RootOf(x**2-3)});
```

$$G := \begin{bmatrix} -\text{RootOf}(-Z^2 - 3) & 0 & 0 & 0 \\ 0 & \text{RootOf}(-Z^2 - 3) & 0 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

```
> op(P);
```

$$\begin{bmatrix} 2 \text{RootOf}(-Z^2 - 3) & -2 \text{RootOf}(-Z^2 - 3) & -3 & 0 \\ -2 & -2 & 0 & -3 \\ -\text{RootOf}(-Z^2 - 3) & \text{RootOf}(-Z^2 - 3) & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

```
> H:=evalm(inverse(P)*A*P):
```

```
> equal(map(simplify,H),G);
```

true

```
> alias(alpha=RootOf(x**2-2)): alias(beta=RootOf(x**2-3)):
```

```
> G:=IRREDUCIBLE(B,Q,{alpha,beta});
```

$$G := \begin{bmatrix} -\alpha & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & -\beta \end{bmatrix}$$

```
> H:=evalm(inverse(Q)*B*Q):
```

```
> equal(map(simplify,H),G);
```

```
true
```

```
> op(Q);
```

$$\begin{bmatrix} 3\alpha & -2\beta & -3\alpha & 2\beta \\ -3 & -2 & -3 & -2 \\ -\alpha & \beta & \alpha & -\beta \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

```
> alias(alpha=RootOf(x**2+2)): alias(beta=RootOf(x**2-3)):
```

```
> G:=IRREDUCIBLE(B,P,{5,beta});
```

$$\begin{bmatrix} 2\alpha & 0 & 0 & 0 \\ 0 & 4\alpha & 0 & 0 \\ 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & 3\alpha \end{bmatrix}$$

```
> op(P);
```

$$\begin{bmatrix} 4\alpha & 2\alpha & 3\alpha & \alpha \\ 2 & 3 & 3 & 2 \\ 2\alpha & 4\alpha & \alpha & 3\alpha \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

```
> H:=evalm(inverse(P)*B*P):
```

```
> M:=modulariza(H,5):
```

```
> equal(M,G);
```

```
true
```

> IRREDUCIBLE(B,inter): # El programa comenzará a ejecutar una sesión interactiva ante el usuario, solicitando a éste datos y opciones relativas al trabajo a realizar.

```
> A:=companion((x-a/b)*(x-c/d),x):
```

```
> G:=IRREDUCIBLE(diag(A,B),P,{5,beta});
```

$$\begin{bmatrix} c/d & 0 & 0 & 0 & 0 & 0 \\ 0 & a/b & 0 & 0 & 0 & 0 \\ 0 & 0 & 4\alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & 3\alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\alpha \end{bmatrix}$$

```
> op(P);
```

$$\begin{bmatrix} 4a/b & 4c/d & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2\alpha & \alpha & 3\alpha & 4\alpha \\ 0 & 0 & 3 & 2 & 3 & 2 \\ 0 & 0 & 4\alpha & 3\alpha & \alpha & 2\alpha \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

```
> evalm(inverse(P)&*diag(A,B)&*P-G):
```

```
> iszero(modulariza(%5));
```

```
true
```

RACIONAL_JORDAN
de una matriz

Calcula la forma Racional_Jordan

Diferentes maneras de utilización:

- RACIONAL_JORDAN(A)
- RACIONAL_JORDAN(A,K)
- RACIONAL_JORDAN(A,P)
- RACIONAL_JORDAN(A,P,K)

Argumentos:

- A - una matriz cuadrada.
- (opcional) P - un símbolo (preferiblemente no “ligado”).
- (opcional) K - un número primo o un conjunto de elementos algebraicos³ sobre $\mathbf{Q}$ o un conjunto formado por un número primo y elementos algebraicos³ sobre el cuerpo primo.

Descripción:

La función **RACIONAL_JORDAN** implementa los procedimientos descritos en [OL2], calculando una matriz $G = \text{diag}[G_1, \dots, G_r]$ semejante a la matriz

³que MAPLE acepte como tales

entrada A , y alojando en la variable P una matriz P tal que $P^{-1}AP = G$, donde

$$G_i = \begin{bmatrix} C(p_i) & E_i & & & & \\ & C(p_i) & E_i & & & \\ & & \ddots & & & \\ & & & C(p_i) & E_i & \\ & & & & C(p_i) & \\ & & & & & C(p_i) \end{bmatrix}$$

siendo p_i un polinomio irreducible de grado m_i , $C(p_i)$ su matriz compañera y E_i la matriz (0) de orden m_i , salvo $E_{1m_i} = 1$

Nótese que para el caso $m_i = 1, \dots, r$ se trata de la forma de Jordan. Para obtener la matriz P , desde el *prompt* de Maple solicítese `op(P)`.

Sea $A = (a_{ij})$.

- `RACIONAL_JORDAN(A)` y `RACIONAL_JORDAN(A,P)` efectúan los cálculos en la extensión $\mathbf{Q}(a_{ij})$.
- Si $K = \{\alpha_1, \dots, \alpha_n\}$ es un conjunto de números algebraicos sobre $\mathbf{Q}$, `RACIONAL_JORDAN(A,K)` y `RACIONAL_JORDAN(A,P,K)` efectúan los cálculos en en la extensión $\mathbf{Q}(a_{ij}, \alpha_1, \dots, \alpha_n)$.
- Si K contiene un elemento algebraico sobre $\mathbf{Z}_p$, `IRREDUCIBLE(A,P,K)` efectúa los cálculos en la extensión $\mathbf{Z}_p(a_{ij})$; debido a que se usa la función `Factors` de MAPLE, los términos a_{ij} de la matriz A pueden incluir a lo sumo un número algebraico en forma de `RootOf`. Para el caso $K = \{p, \alpha\}$, el ordenador puede tomarse su tiempo y memoria. Teste el usuario la matriz

$$\begin{bmatrix} a/b & c/d \\ e/f & g/h \end{bmatrix} \text{ sobre } \mathbf{Z}_5(\alpha) \quad \alpha^2 - 2 = 0$$

Ejemplos:

```
> B:=companion((x**2-a)**2,x):
```

```
> equal(IRREDUCIBLE(B),B);
```

```
true
```

Sin embargo,

```
> G:=RACIONAL_JORDAN(B);
```

$$\begin{bmatrix} 0 & a & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & a \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

> G:=RACIONAL_JORDAN(B,P):

> op(P);

$$\begin{bmatrix} -a & 0 & 1 & 0 \\ 0 & -a & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

> H:=evalm(inverse(P)&*B&*P):

> equal(map(simplify,H),G);

true

> a:=2: G:=RACIONAL_JORDAN(B,P,{sqrt(a)});

$$G := \begin{bmatrix} \sqrt{2} & 1 & 0 & 0 \\ 0 & \sqrt{2} & 0 & 0 \\ 0 & 0 & -\sqrt{2} & 1 \\ 0 & 0 & 0 & -\sqrt{2} \end{bmatrix}$$

> op(P);

$$\begin{bmatrix} -2\sqrt{2} & 2 & 2\sqrt{2} & 2 \\ -2 & 2\sqrt{2} & -2 & -2\sqrt{2} \\ \sqrt{2} & 1 & -\sqrt{2} & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

> H:=evalm(inverse(P)&*B&*P):

> equal(map(simplify,H),G);

true

> B:=companion((x**2-2)**2*(x**2-5),x):

> G:=RACIONAL_JORDAN(B,P,3);

$$G := \begin{bmatrix} 0 & 2 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

> op(P);

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

> alias(alpha=RootOf(x**2-2)): alias(beta=RootOf(x**2+3)):

> G:=RACIONAL_JORDAN(B,P,{5,alpha});

$$G := \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \beta & 1 & 0 & 0 \\ 0 & 0 & 0 & \beta & 0 & 0 \\ 0 & 0 & 0 & 0 & 4\beta & 1 \\ 0 & 0 & 0 & 0 & 0 & 4\beta \end{bmatrix}$$

> op(P);

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

> H:=evalm(inverse(P)*B*P):

```
> M:=modulariza(H,5);  
  
> equal(M,G);  
true
```

4.2 Otros procedimientos de interés del archivo Jordan

PILAR **Calcula el pilar asociado a un factor o un valor propio**

Diferentes formas de utilización:

- `PILAR(f,g,x)`
- `PILAR(f,g,x,K)`
- `PILAR(t,m,f,x)`
- `PILAR(t,m,f,x,K)`

Argumentos:

- x - una indeterminada
- (opcional) K - un cuerpo
- f - un polinomio, usualmente mónico, en $L[x]$, donde L es la extensión de K que resulta de adjuntar los coeficientes de f
- g - un factor de f en $L[x]$
- t ($\in K$) un valor propio de f
- m la multiplicidad de t

Si el parámetro opcional no se incluye el programa toma, por defecto, el cuerpo de los números racionales.

Descripción:

La función `PILAR` implementa las definiciones 5.1, p.111 y 6.4, p.118 del libro, básicas en el algoritmo que construye la matriz de cambio en las formas canónicas irreducible y de Jordan de una matriz.

PILAR(f,g,x) y PILAR(f,g,x,K) dan el pilar asociado al factor g de f, mientras que PILAR(t,m,f,x) y PILAR(t,m,f,x,K) implementan el concepto de pilar asociado al valor propio t.

El parámetro K puede ser bien un número primo p , bien un conjunto formado por un primo p y un (único) elemento algebraico sobre $\mathbf{Z}_p$, bien un conjunto $\{\alpha_1, \dots, \alpha_n\}$ de números algebraicos.

Deben observarse las advertencias hechas para el caso de característica prima en la descripción de la función IRREDUCIBLE.

Ejemplos:

> f:=(x-3)**2*(x**2-2):

> g:=(x-3)**2:

> PILAR(f,g,x);

$$\begin{bmatrix} -2 & 0 \\ 0 & -2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

> PILAR(f,g,x,3);

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

> g:=x-sqrt(2):

> PILAR(f,g,x,{sqrt(2)});

$$\begin{bmatrix} 9\sqrt{2} \\ 9-6\sqrt{2} \\ -6+\sqrt{2} \\ 1 \end{bmatrix}$$

> PILAR(3,2,f,x);

$$\begin{bmatrix} -2 & 6 \\ 0 & -2 \\ 1 & -3 \\ 0 & 1 \end{bmatrix}$$

> PILAR(-sqrt(2),1,f,x,{sqrt(2)});

$$\begin{bmatrix} -9\sqrt{2} \\ 9 + 6\sqrt{2} \\ -6 - \sqrt{2} \\ 1 \end{bmatrix}$$

> f:=(x-3*a)**2*(x**2-b):

> g:=(x-3*a)**2:

> PILAR(f,g,x);

$$\begin{bmatrix} -b & 0 \\ 0 & -b \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

> PILAR(f,g,x,3);

$$\begin{bmatrix} 2b & 0 \\ 0 & 2b \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

> g:=x-sqrt(b):

> PILAR(f,g,x,{sqrt(b)});

$$\begin{bmatrix} 9a^2\sqrt{b} \\ 9a^2 - 6a\sqrt{b} \\ -6a + \sqrt{b} \\ 1 \end{bmatrix}$$

> PILAR(3*a,2,f,x);

$$\begin{bmatrix} -b & 3ab \\ 0 & -b \\ 1 & -3a \\ 0 & 1 \end{bmatrix}$$

> PILAR(-sqrt(b),1,f,x,{sqrt(b)});

$$\begin{bmatrix} -9a^2\sqrt{b} \\ 9a^2 + 6a\sqrt{b} \\ -6a - \sqrt{b} \\ 1 \end{bmatrix}$$

t_PILAR **Calcula el (t,p,m)-pilar de un polinomio**

Diferentes formas de utilización:

- `t_PILAR(t,p,m,f,x)`
- `t_PILAR(t,p,m,f,x,K)`

Argumentos:

- `x` - una indeterminada
- (opcional) `K` - un cuerpo
- `f` - un polinomio, usualmente mónico, en $L[x]$, donde L es la extensión de K que resulta de adjuntar los coeficientes de `f`
- `p` - un factor de `f` en $L[x]$
- `t` un elemento de K
- `m` un número natural no superior al grado de `p`

Si el parámetro opcional no se incluye el programa toma, por defecto, el cuerpo de los números racionales.

Descripción:

La función `t_PILAR`, versión unificada de la función `PILAR`, implementa la definición 2.1 del artículo

Unified method for determining canonical forms of a matrix
ACM SIGSAM Bulletin Vol. 33, No. 1 March 1999

Tal concepto es básico en los algoritmos que construyen la matriz de cambio en las formas canónicas irreducible, de Jordan y Racional_Jordan de una matriz.

El usuario elegirá entre una función u otra.

El parámetro `K` puede ser bien un número primo p , bien un conjunto formado por un primo p y un (único) elemento algebraico sobre $\mathbf{Z}_p$, bien un conjunto $\{\alpha_1, \dots, \alpha_n\}$ de números algebraicos.

Deben observarse las advertencias hechas para el caso de característica prima en la descripción de la función `IRREDUCIBLE`.

Si el parámetro m supera el grado de p , el programa dará el correspondiente mensaje de ERROR.

Ejemplos:

> f:=(x-3*a)**2*(x**2-b):

> p:=(x-3*a)**2:

> t_PILAR(0,p,2,f,x);

$$\begin{bmatrix} -b & 0 \\ 0 & -b \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

> p:=x-sqrt(b):

> t_PILAR(0,p,1,f,x,{sqrt(b)});

$$\begin{bmatrix} 9a^2\sqrt{b} \\ 9a^2 - 6a\sqrt{b} \\ -6a + \sqrt{b} \\ 1 \end{bmatrix}$$

> t_PILAR(3*a,(x-3*a)**2,2,f,x);

$$\begin{bmatrix} -b & 3ab \\ 0 & -b \\ 1 & -3a \\ 0 & 1 \end{bmatrix}$$

> t_PILAR(-sqrt(b),x**2-b,1,f,x,{sqrt(b)});

$$\begin{bmatrix} 9a^2 \\ -6a \\ 1 \\ 0 \end{bmatrix}$$

> t_PILAR(-sqrt(b),x**2-b,2,f,x,{sqrt(b)});

$$\begin{bmatrix} 9a^2 & 9a^2\sqrt{b} \\ -6a & 9a^2 - 6a\sqrt{b} \\ 1 & -6a + \sqrt{b} \\ 0 & 1 \end{bmatrix}$$

LJordan_compa**Procedimiento interactivo
para la forma de Jordan
de una matriz compañera**

Utilización:

LJordan_compa(C)

Argumento:

C - una matriz compañera

Descripción:

La función LJordan_compa implementa interactivamente los procedimientos de la página 121 del libro para calcular la forma canónica de Jordan de una matriz compañera. Su utilización se aconseja antes de aprender a calcular la forma de Jordan de una matriz general.

Ejemplos:

```
> C:=companion((x2 - 4)2):
```

```
> LJordan_compa(C); # El programa comenzará a ejecutar una sesión interactiva ante el usuario, solicitando a éste datos y opciones relativas al trabajo a realizar.
```