16.42 NORMFORM: Computation of matrix normal forms

This package contains routines for computing the following normal forms of matrices:

- smithex_int
- smithex
- frobenius
- ratjordan
- jordansymbolic
- jordan.

Author: Matt Rebbeck.

16.42.1 Introduction

When are two given matrices similar? Similar matrices have the same trace, determinant, characteristic polynomial, and eigenvalues, but the matrices

$$\mathcal{U} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad \mathcal{V} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

are the same in all four of the above but are not similar. Otherwise there could exist a nonsingular $\mathcal{N} \in M_2$ (the set of all 2×2 matrices) such that $\mathcal{U} = \mathcal{N}\mathcal{V}\mathcal{N}^{-1} = \mathcal{N}\,\mathit{0}\,\mathcal{N}^{-1} = \mathit{0}$, which is a contradiction since $\mathcal{U} \neq \mathit{0}$.

Two matrices can look very different but still be similar. One approach to determining whether two given matrices are similar is to compute the normal form of them. If both matrices reduce to the same normal form they must be similar.

NORMFORM is a package for computing the following normal forms of matrices:

```
 - smithex
 - smithex_int
 - frobenius
 - ratjordan
 - jordansymbolic
 - jordan
```

The package is loaded by `load_package normform;`

By default all calculations are carried out in $\mathbf{Q}$ (the rational numbers). For `smithex`, `frobenius`, `ratjordan`, `jordansymbolic`, and `jordan`, this field can be extended. Details are given in the respective sections.

The `frobenius`, `ratjordan`, and `jordansymbolic` normal forms can also be computed in a modular base. Again, details are given in the respective sections.

The algorithms for each routine are contained in the source code.

NORMFORM has been converted from the normform and Normform packages written by T.M.L. Mulders and A.H.M. Levelt. These have been implemented in Maple [4].

16.42.2 Smith normal form

Function

`smithex`($\mathcal{A}$, x) computes the Smith normal form $\mathcal{S}$ of the matrix $\mathcal{A}$.

It returns $\{\mathcal{S}, \mathcal{P}, \mathcal{P}^{-1}\}$ where $\mathcal{S}, \mathcal{P}$, and $\mathcal{P}^{-1}$ are such that $\mathcal{P}\mathcal{S}\mathcal{P}^{-1} = \mathcal{A}$.

$\mathcal{A}$ is a rectangular matrix of univariate polynomials in x.

x is the variable name.

Field extensions

Calculations are performed in $\mathbf{Q}$. To extend this field the ARNUM package can be used. For details see subsection 16.42.8.

Synopsis:

- The Smith normal form $\mathcal{S}$ of an n by m matrix $\mathcal{A}$ with univariate polynomial entries in x over a field $\mathbf{F}$ is computed. That is, the polynomials are then regarded as elements of the *E*uclidean domain $\mathbf{F}(x)$.
- The Smith normal form is a diagonal matrix $\mathcal{S}$ where:
  - $\operatorname{rank}(\mathcal{A})$ = number of nonzero rows (columns) of $\mathcal{S}$.
  - $\mathcal{S}(i,i)$ is a monic polynomial for $0 < i \leq \operatorname{rank}(\mathcal{A})$.
  - $\mathcal{S}(i,i)$ divides $\mathcal{S}(i+1,i+1)$ for $0 < i < \operatorname{rank}(\mathcal{A})$.
  - $\mathcal{S}(i,i)$ is the greatest common divisor of all i by i minors of $\mathcal{A}$.

  Hence, if we have the case that $n = m$, as well as $\operatorname{rank}(\mathcal{A}) = n$, then

$$\prod_{i=1}^{n} \mathcal{S}(i,i) = \frac{\det(\mathcal{A})}{\operatorname{lcoeff}(\det(\mathcal{A}), x)}.$$

- The Smith normal form is obtained by doing elementary row and column operations. This includes interchanging rows (columns), multiplying through a row (column) by -1, and adding integral multiples of one row (column) to another.

- Although the rank and determinant can be easily obtained from $\mathcal{S}$, this is not an efficient method for computing these quantities except that this may yield a partial factorization of $\det(\mathcal{A})$ without doing any explicit factorizations.

Example:

```
load_package normform;
```

$$\mathcal{A} = \begin{pmatrix} x & x+1 \\ 0 & 3*x^2 \end{pmatrix}$$

$$\mathtt{smithex}(\mathcal{A}, x) = \left\{ \begin{pmatrix} 1 & 0 \\ 0 & x^3 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 3*x^2 & 1 \end{pmatrix}, \begin{pmatrix} x & x+1 \\ -3 & -3 \end{pmatrix} \right\}$$

16.42.3 smithex_int

Function

Given an n by m rectangular matrix $\mathcal{A}$ that contains *only* integer entries, `smithex_int`($\mathcal{A}$) computes the Smith normal form $\mathcal{S}$ of $\mathcal{A}$.

It returns $\{\mathcal{S}, \mathcal{P}, \mathcal{P}^{-1}\}$ where $\mathcal{S}$, $\mathcal{P}$, and $\mathcal{P}^{-1}$ are such that $\mathcal{P}\mathcal{S}\mathcal{P}^{-1} = \mathcal{A}$.

Synopsis

- The Smith normal form $\mathcal{S}$ of an n by m matrix $\mathcal{A}$ with integer entries is computed.
- The Smith normal form is a diagonal matrix $\mathcal{S}$ where:
  - $\mathrm{rank}(\mathcal{A})$ = number of nonzero rows (columns) of $\mathcal{S}$.
  - $\mathrm{sign}(\mathcal{S}(i,i)) = 1$ for $0 < i \le \mathrm{rank}(\mathcal{A})$.
  - $\mathcal{S}(i,i)$ divides $\mathcal{S}(i+1,i+1)$ for $0 < i < \mathrm{rank}(\mathcal{A})$.
  - $\mathcal{S}(i,i)$ is the greatest common divisor of all i by i minors of $\mathcal{A}$.

  Hence, if we have the case that $n = m$, as well as rank($\mathcal{A}$) $= n$, then

  $$|\det(\mathcal{A})| = \prod_{i=1}^{n} \mathcal{S}(i,i).$$

- The Smith normal form is obtained by doing elementary row and column operations. This includes interchanging rows (columns), multiplying through a row (column) by -1, and adding integral multiples of one row (column) to another.

Example

```
load_package normform;
```

$$\mathcal{A} = \begin{pmatrix} 9 & -36 & 30 \\ -36 & 192 & -180 \\ 30 & -180 & 180 \end{pmatrix}$$

$$\texttt{smithex\_int}(\mathcal{A}) = \left\{ \begin{pmatrix} 3 & 0 & 0 \\ 0 & 12 & 0 \\ 0 & 0 & 60 \end{pmatrix}, \begin{pmatrix} -17 & -5 & -4 \\ 64 & 19 & 15 \\ -50 & -15 & -12 \end{pmatrix}, \begin{pmatrix} 1 & -24 & 30 \\ -1 & 25 & -30 \\ 0 & -1 & 1 \end{pmatrix} \right\}$$

16.42.4 frobenius

Function

`frobenius`$(\mathcal{A})$ computes the Frobenius normal form $\mathcal{F}$ of the matrix $\mathcal{A}$.

It returns $\{\mathcal{F}, \mathcal{P}, \mathcal{P}^{-1}\}$ where $\mathcal{F}$, $\mathcal{P}$, and $\mathcal{P}^{-1}$ are such that $\mathcal{P}\mathcal{F}\mathcal{P}^{-1} = \mathcal{A}$.

$\mathcal{A}$ is a square matrix.

Field extensions

Calculations are performed in $\mathcal{Q}$. To extend this field the ARNUM package can be used. For details see subsection 16.42.8

Modular arithmetic

`frobenius` can be calculated in a modular base. For details see subsection 16.42.9.

Synopsis

- $\mathcal{F}$ has the following structure:

$$\mathcal{F} = \begin{pmatrix} \mathcal{C}p_1 & & & \\ & \mathcal{C}p_2 & & \\ & & \ddots & \\ & & & \mathcal{C}p_k \end{pmatrix}$$

  where the $\mathcal{C}(p_i)$'s are companion matrices associated with polynomials $p_1, p_2, \ldots, p_k$, with the property that p_i divides p_{i+1} for $i = 1 \ldots k-1$. All unmarked entries are zero.

- The Frobenius normal form defined in this way is unique (ie: if we require that p_i divides p_{i+1} as above).

Example

```
load_package normform;
```

$$\mathcal{A} = \begin{pmatrix} \frac{-x^2+y^2+y}{y} & \frac{-x^2+x+y^2-y}{y} \\ \frac{-x^2-x+y^2+y}{y} & \frac{-x^2+x+y^2-y}{y} \end{pmatrix}$$

`frobenius`$(\mathcal{A}) =$

$$\left\{ \begin{pmatrix} 0 & \frac{x*(x^2-x-y^2+y)}{y} \\ 1 & \frac{-2*x^2+x+2*y^2}{y} \end{pmatrix}, \begin{pmatrix} 1 & \frac{-x^2+y^2+y}{y} \\ 0 & \frac{-x^2-x+y^2+y}{y} \end{pmatrix}, \begin{pmatrix} 1 & \frac{-x^2+y^2+y}{x^2+x-y^2-y} \\ 0 & \frac{-y}{x^2+x-y^2-y} \end{pmatrix} \right\}$$

16.42.5 ratjordan

Function

`ratjordan`$(\mathcal{A})$ computes the rational Jordan normal form $\mathcal{R}$ of the matrix $\mathcal{A}$.

It returns $\{\mathcal{R}, \mathcal{P}, \mathcal{P}^{-1}\}$ where $\mathcal{R}$, $\mathcal{P}$, and $\mathcal{P}^{-1}$ are such that $\mathcal{P}\mathcal{R}\mathcal{P}^{-1} = \mathcal{A}$.

$\mathcal{A}$ is a square matrix.

Field extensions

Calculations are performed in $\mathcal{Q}$. To extend this field the ARNUM package can be used. For details see subsection 16.42.8.

Modular arithmetic

`ratjordan` can be calculated in a modular base. For details see subsection 16.42.9.

Synopsis

- $\mathcal{R}$ has the following structure:

$$\mathcal{R} = \begin{pmatrix} r_{11} & & & & & \\ & r_{12} & & & & \\ & & \ddots & & & \\ & & & r_{21} & & \\ & & & & r_{22} & \\ & & & & & \ddots \end{pmatrix}$$

The r_{ij}'s have the following shape:

$$r_{ij} = \begin{pmatrix} \mathcal{C}(p) & \mathcal{I} & & & \\ & \mathcal{C}(p) & \mathcal{I} & & \\ & & \ddots & \ddots & \\ & & & \mathcal{C}(p) & \mathcal{I} \\ & & & & \mathcal{C}(p) \end{pmatrix}$$

where there are eij times $\mathcal{C}(p)$ blocks along the diagonal and $\mathcal{C}(p)$ is the companion matrix associated with the irreducible polynomial p. All unmarked entries are zero.

Example

```
load_package normform;
```

$$\mathcal{A} = \begin{pmatrix} x+y & 5 \\ y & x^2 \end{pmatrix}$$

`ratjordan`$(\mathcal{A}) =$

$$\left\{ \begin{pmatrix} 0 & -x^3 - x^2 * y + 5 * y \\ 1 & x^2 + x + y \end{pmatrix}, \begin{pmatrix} 1 & x+y \\ 0 & y \end{pmatrix}, \begin{pmatrix} 1 & \frac{-(x+y)}{y} \\ 0 & \frac{1}{y} \end{pmatrix} \right\}$$

16.42.6 jordansymbolic

Function

`jordansymbolic`$(\mathcal{A})$ computes the Jordan normal form $\mathcal{J}$ of the matrix $\mathcal{A}$.

It returns $\{\mathcal{J}, \mathcal{L}, \mathcal{P}, \mathcal{P}^{-1}\}$, where $\mathcal{J}$, $\mathcal{P}$, and $\mathcal{P}^{-1}$ are such that $\mathcal{P}\mathcal{J}\mathcal{P}^{-1} = \mathcal{A}$. $\mathcal{L} = \{ll, \xi\}$, where ξ is a name and ll is a list of irreducible factors of $p(\xi)$.

$\mathcal{A}$ is a square matrix.

Field extensions

Calculations are performed in $\mathcal{Q}$. To extend this field the ARNUM package can be used. For details see subsection 16.42.8.

Modular arithmetic

`jordansymbolic` can be calculated in a modular base. For details see subsection 16.42.9.

Extras

If using `xr`, the X interface for REDUCE, the appearance of the output can be improved by setting the switch `looking_good` to on. This converts all lambda to ξ and improves the indexing, e.g., `lambda12` $\Rightarrow \xi_{12}$. The example below shows the output when this switch is on.

Synopsis

- A *Jordan block* $\jmath_k(\lambda)$ is a k by k upper triangular matrix of the form:

$$\jmath_k(\lambda) = \begin{pmatrix} \lambda & 1 & & & \\ & \lambda & 1 & & \\ & & \ddots & \ddots & \\ & & & \lambda & 1 \\ & & & & \lambda \end{pmatrix}$$

  There are $k-1$ terms "$+1$" in the superdiagonal; the scalar λ appears k times on the main diagonal. All other matrix entries are zero, and $\jmath_1(\lambda) = (\lambda)$.

- A Jordan matrix $\mathcal{J} \in M_n$ (the set of all n by n matrices) is a direct sum of *jordan blocks*

$$\mathcal{J} = \begin{pmatrix} \jmath_{n_1}(\lambda_1) & & & \\ & \jmath_{n_2}(\lambda_2) & & \\ & & \ddots & \\ & & & \jmath_{n_k}(\lambda_k) \end{pmatrix}, n_1 + n_2 + \cdots + n_k = n$$

  in which the orders n_i may not be distinct and the values λ_i need not be distinct.

- Here λ is a zero of the characteristic polynomial p of $\mathcal{A}$. If p does not split completely, symbolic names are chosen for the missing zeroes of p. If, by some means, one knows such missing zeroes, they can be substituted for the symbolic names. For this, `jordansymbolic` actually returns $\{\mathcal{J}, \mathcal{L}, \mathcal{P}, \mathcal{P}^{-1}\}$. $\mathcal{J}$ is the Jordan normal form of $\mathcal{A}$ (using symbolic names if necessary). $\mathcal{L} = \{ll, \xi\}$, where ξ is a name and ll is a list of irreducible factors of $p(\xi)$. If symbolic names are used then ξ_{ij} is a zero of ll_i. $\mathcal{P}$ and $\mathcal{P}^{-1}$ are as above.

Example

```
load_package normform;
on looking_good;
```

$$\mathcal{A} = \begin{pmatrix} 1 & y \\ y^2 & 3 \end{pmatrix}$$

`jordansymbolic`$(\mathcal{A}) =$

$$\left\{ \begin{pmatrix} \xi_{11} & 0 \\ 0 & \xi_{12} \end{pmatrix}, \left\{ \left\{ -y^3 + \xi^2 - 4 * \xi + 3 \right\}, \xi \right\}, \begin{pmatrix} \xi_{11} - 3 & \xi_{12} - 3 \\ y^2 & y^2 \end{pmatrix}, \begin{pmatrix} \frac{\xi_{11}-2}{2*(y^3-1)} & \frac{\xi_{11}+y^3-1}{2*y^2*(y^3+1)} \\ \frac{\xi_{12}-2}{2*(y^3-1)} & \frac{\xi_{12}+y^3-1}{2*y^2*(y^3+1)} \end{pmatrix} \right\}$$

```
solve(-y^3+xi^2-4*xi+3,xi);
```

$$\{\xi = \sqrt{y^3+1} + 2, \xi = -\sqrt{y^3+1} + 2\}$$

```
J = sub({xi(1,1)=sqrt(y^3+1)+2, xi(1,2)=-sqrt(y^3+1)+2},
        first jordansymbolic (A))
```

$$\mathcal{J} = \begin{pmatrix} \sqrt{y^3+1} + 2 & 0 \\ 0 & -\sqrt{y^3+1} + 2 \end{pmatrix}$$

For a similar example ot this in standard REDUCE (ie: not using `xr`), see the `normform.rlg` file.

16.42.7 jordan

Function

`jordan`$(\mathcal{A})$ computes the Jordan normal form $\mathcal{J}$ of the matrix $\mathcal{A}$.

It returns $\{\mathcal{J}, \mathcal{P}, \mathcal{P}^{-1}\}$, where $\mathcal{J}$, $\mathcal{P}$, and $\mathcal{P}^{-1}$ are such that $\mathcal{P}\mathcal{J}\mathcal{P}^{-1} = \mathcal{A}$.

$\mathcal{A}$ is a square matrix.

Field extensions

Calculations are performed in $\mathcal{Q}$. To extend this field the ARNUM package can be used. For details see subsection 16.42.8.

Note

In certain polynomial cases the switch `fullroots` is turned on to compute the zeroes. This can lead to the calculation taking a long time, as well as the output being very large. In this case a message
`***** WARNING: fullroots turned on. May take a while.`
will be printed. It may be better to kill the calculation and compute `jordansymbolic` instead.

Synopsis

- The Jordan normal form $\mathcal{J}$ with entries in an algebraic extension of $\mathcal{Q}$ is computed.
- A *Jordan block* $\jmath_k(\lambda)$ is a k by k upper triangular matrix of the form:

$$\jmath_k(\lambda) = \begin{pmatrix} \lambda & 1 & & & \\ & \lambda & 1 & & \\ & & \ddots & \ddots & \\ & & & \lambda & 1 \\ & & & & \lambda \end{pmatrix}$$

  There are $k-1$ terms "$+1$" in the superdiagonal; the scalar λ appears k times on the main diagonal. All other matrix entries are zero, and $\jmath_1(\lambda) = (\lambda)$.
- A Jordan matrix $\mathcal{J} \in M_n$ (the set of all n by n matrices) is a direct sum of *jordan blocks*.

$$\mathcal{J} = \begin{pmatrix} \jmath_{n_1}(\lambda_1) & & & \\ & \jmath_{n_2}(\lambda_2) & & \\ & & \ddots & \\ & & & \jmath_{n_k}(\lambda_k) \end{pmatrix}, n_1 + n_2 + \cdots + n_k = n$$

  in which the orders n_i may not be distinct and the values λ_i need not be distinct.
- Here λ is a zero of the characteristic polynomial p of $\mathcal{A}$. The zeroes of the characteristic polynomial are computed exactly, if possible. Otherwise they are approximated by floating point numbers.

Example

```
load_package normform;
```

$$\mathcal{A} = \begin{pmatrix} -9 & -21 & -15 & 4 & 2 & 0 \\ -10 & 21 & -14 & 4 & 2 & 0 \\ -8 & 16 & -11 & 4 & 2 & 0 \\ -6 & 12 & -9 & 3 & 3 & 0 \\ -4 & 8 & -6 & 0 & 5 & 0 \\ -2 & 4 & -3 & 0 & 1 & 3 \end{pmatrix}$$

```
J = first jordan(A);
```

$$\mathcal{J} = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & i+2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -i+2 \end{pmatrix}$$

16.42.8 Algebraic extensions: Using the ARNUM package

The package is loaded by the command `load_package arnum;`. The algebraic field **Q** can now be extended. For example, `defpoly sqrt2**2-2;` will extend it to include $\sqrt{2}$ (defined here by `sqrt2`). The ARNUM package was written by Eberhard Schrüfer and is described in section 16.3.

16.42.8.1 Example

```
load_package normform;
load_package arnum;
defpoly sqrt2**2-2;
```

(sqrt2 now changed to $\sqrt{2}$ for looks!)

$$\mathcal{A} = \begin{pmatrix} 4*\sqrt{2}-6 & -4*\sqrt{2}+7 & -3*\sqrt{2}+6 \\ 3*\sqrt{2}-6 & -3*\sqrt{2}+7 & -3*\sqrt{2}+6 \\ 3*\sqrt{2} & 1-3*\sqrt{2} & -2*\sqrt{2} \end{pmatrix}$$

$$\texttt{ratjordan}(\mathcal{A}) = \left\{ \begin{pmatrix} \sqrt{2} & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ 0 & 0 & -3*\sqrt{2}+1 \end{pmatrix}, \right.$$

$$\begin{pmatrix} 7*\sqrt{2}-6 & \frac{2*\sqrt{2}-49}{31} & \frac{-21*\sqrt{2}+18}{31} \\ 3*\sqrt{2}-6 & \frac{21*\sqrt{2}-18}{31} & \frac{-21*\sqrt{2}+18}{31} \\ 3*\sqrt{2}+1 & \frac{-3*\sqrt{2}+24}{31} & \frac{3*\sqrt{2}-24}{31} \end{pmatrix},$$

$$\left. \begin{pmatrix} 0 & \sqrt{2}+1 & 1 \\ -1 & 4*\sqrt{2}+9 & 4*\sqrt{2} \\ -1 & -\frac{1}{6}*\sqrt{2}+1 & 1 \end{pmatrix} \right\}$$

16.42.9 Modular arithmetic

Calculations can be performed in a modular base by setting the switch `modular` to on. The base can then be set by `setmod p;` (p a prime). The normal form will then have entries in $\mathcal{Z}/\mathrm{p}\mathcal{Z}$.

By also switching on `balanced_mod` the output will be shown using a symmetric modular representation.

Information on this modular manipulation can be found in chapter 9.

16.42.9.1 Example

```
load_package normform;
on modular;
setmod 23;
```

$$\mathcal{A} = \begin{pmatrix} 10 & 18 \\ 17 & 20 \end{pmatrix}$$

$\texttt{jordansymbolic}(\mathcal{A}) =$

$$\left\{ \begin{pmatrix} 18 & 0 \\ 0 & 12 \end{pmatrix}, \{\{\lambda+5, \lambda+11\}, \lambda\}, \begin{pmatrix} 15 & 9 \\ 22 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 14 \\ 1 & 15 \end{pmatrix} \right\}$$

```
on balanced_mod;
```

$\texttt{jordansymbolic}(\mathcal{A}) =$

$$\left\{ \begin{pmatrix} -5 & 0 \\ 0 & -11 \end{pmatrix}, \{\{\lambda + 5, \lambda + 11\}, \lambda\}, \begin{pmatrix} -8 & 9 \\ -1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & -9 \\ 1 & -8 \end{pmatrix} \right\}$$

Bibliography

[1] T.M.L.Mulders and A.H.M. Levelt: *The Maple normform and Normform packages.* (1993)

[2] T.M.L.Mulders: *Algoritmen in De Algebra, A Seminar on Algebraic Algorithms, Nigmegen.* (1993)

[3] Roger A. Horn and Charles A. Johnson: *Matrix Analysis.* Cambridge University Press (1990)

[4] Bruce W. Chat... [et al.]: *Maple (Computer Program).* Springer-Verlag (1991)

[5] Anthony C. Hearn: REDUCE *User's Manual 3.6.* RAND (1995)