Recursive matrix algorithms in rings and semirings

G. Malaschonok
Tambov State University
malaschonok@gmail.com

Computer Algebra Seminar
MSU, Moscow

04.23.2014

Introduction

The report focuses on recursive matrix algorithms, which are used in computer algebra: matrix inversion , the calculation of the matrix closure, a triangular decomposition and others. Recursive block matrix algorithms have two advantages. For dense matrices , these algorithms have the complexity of matrix multiplication. Sparse matrix has an additional advantage: fill factor matrix has a moderate growth compared to standard algorithms such as Gauss or LU decomposition.
The report will illustrate the similarities between the algorithm of calculation of matrix closures and matrix inversion algorithm.

The second part of the report is devoted to presenting a web service MathPartner.

Math Partner is the first CAaaS system (Computer Algebra as a Servise). It is hosted at mathpar.com. The reports provides examples of solutions specific analytical problems, possibilities of 2D and 3D graphics and more. The user language of this service is similar to the language TEXand expanded by control operators and by operators of creating procedures.

Interest of algebraists in the application of this service may be caused by the possibility to perform calculations not only in the domain $\mathbb{Z}$ or $\mathbb{Q}$, in the field approximation $\mathbb{R}$ or $\mathbb{C}$, but in a finite fields $\mathbb{Z}_p$, in 18 different tropical algebras (semirings and semifields), as well as in the rings of polynomials over them. User can compute the Gröbner basis of polynomial ideal. This allows him to perform calculations in the factor ring.

Environment for mathematical objects

To select the environment you have to set the *algebraic structure*.

By default, a space of the four real variables is defined

$$\mathbb{R}64[x, y, z, t].$$

This is ring of polynomials with coefficients in the ring of real numbers.
The variables are arranged in order from left to right: $x < y < z < t$

User can change the environment:
For example the space

$$\mathbb{Q}[x, y, z]$$

may be suitable to solve many problems of school mathematics.

The installation command should be the follow: SPACE = Q [x, y, z];
Moving a mathematical object from the previous environment to the current environment, as a rule, should be performed explicitly, using the function

$$toNewRing()$$

In some cases, such a transformation to the current environment is automatic.

All other names which are not listed as a variables can be chosen arbitrarily by the user for any mathematical object.
For example

$$a = x + 1, \quad f = \backslash sin(x + y) - a.$$

The rule:

If the object name begins with a *capital letter* such object is an element of a *noncommutative* algebra.
If the object name begins with a *lowercase letter* such object is an element of a *commutative* algebra.

Numerical sets with standard operations

Z — the set of integers $\mathbb{Z}$,
Zp — a finite field $\mathbb{Z}/p\mathbb{Z}$ where p is a prime number,
Zp32 — a finite field $\mathbb{Z}/p\mathbb{Z}$ where p is less 2^{31},
Z64 — the ring of integer numbers z such that $-2^{63} \leqslant z < 2^{63}$,
Q — the set of rational numbers,
R — approximate real numbers with arbitrary mantissa,
R64 — standard floating-point 64-bit numbers
R128 — floating-point 64-bit numbers, equipped 64-bit for the order,
C — complexification of R,
C64 — complexification of R64,
C128 — complexification of R128,
CZ — complexification of of Z,
CZp — complexification of Zp,
CZp32 — complexification of Zp32,
CZ64 — complexification of Z64,
CQ — complexification of Q.

Examples of simple commutative polynomial rings:
SPACE = Z [x, y, z];
SPACE = R64 [u, v];
SPACE = C [x].

Constants

ACCURACY — an amount of exact decimal positions in the fractional part of a real numbers of type R in the result of multiplication or division operation.
FLOATPOS —an amount of decimal positions of the real number of type R or R64, which you can see in the printed form.
ZERO_R — a machine zero for R and C numbers.
ZERO_R64 — a machine zero for R64, R128, C64 and C128 numbers.
MOD32 — the module for a finite field of the type Zp32, its value is not greater than 2^{31}.
MOD — the module for a finite field of the type Zp.
To set the machine zero $1/10^9$ (i.e. $1E-9$), you can use the commands $ZERO\_R = 9$ or $ZERO\_R64 = 9$.

Example.

```
SPACE=Zp32[x, y];
MOD32=7;
f=37x+42y+55;
g=2*f;
\print(f , g );
```

The results:

f = 2x-1;
g = 4x+5.

Idempotent algebra and tropical mathematics

User can uses the idempotent algebras. In this case the signs of "addition"and "multiplication"for the infix operations can be used for operations in tropical algebra: min, max, addition, multiplication.
Each numerical sets $\mathbb{R}$, $\mathbb{R}64$, $\mathbb{Z}$ has two additional elements ∞ and $-\infty$, and they have different elements, which is play the role of zero and unit. We denote these sets $\hat{\mathbb{R}}$, $\hat{\mathbb{R}}64$, $\hat{\mathbb{Z}}$, correspondingly. The name of tropical algebra is obtained from three words: (1) a numerical set, (2) an operation, which corresponding to the sign *plus* and (3) an operation, which corresponding to the sign *times*.
The algebras *R64MaxPlus*, *R64MinPlus*, *R64MaxMin*, *R64MinMax*, *R64MaxMult*, *R64MinMult* are defined for the numerical set $\hat{\mathbb{R}}64$.
RMaxPlus, RMinPlus, RMaxMin, R64MinMax, RMaxMult, RMinMult are defined for the numerical set $\hat{\mathbb{R}}$.
ZMaxPlus, ZMinPlus, ZMaxMin, ZMinMax, ZMaxMult, ZMinMult are defined for the numerical set $\hat{\mathbb{Z}}$.

For example, for the algebra *ZMaxPlus* you can do the following operations.

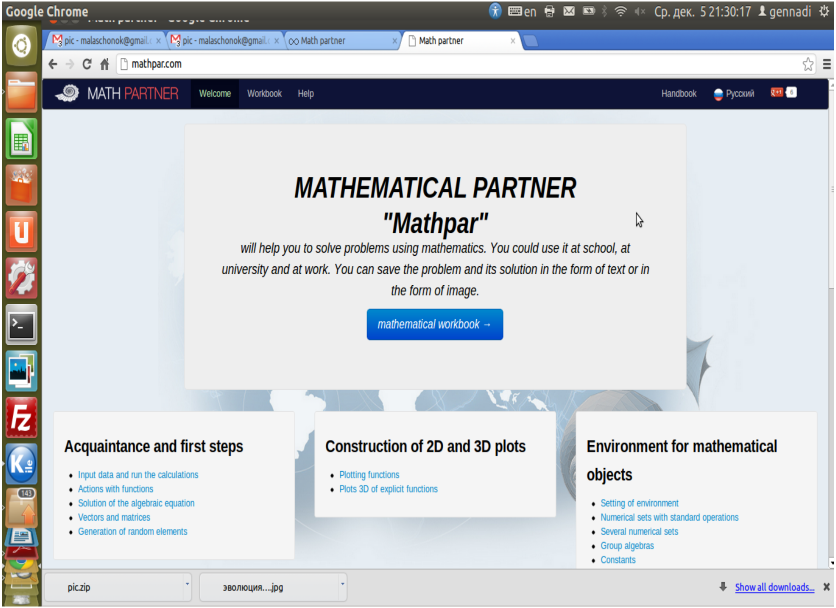
Example.

```
SPACE=ZMaxPlus[x, y];
a=2; b=9+x; c=a+b; d=a*b+y;
\print(c, d);
```

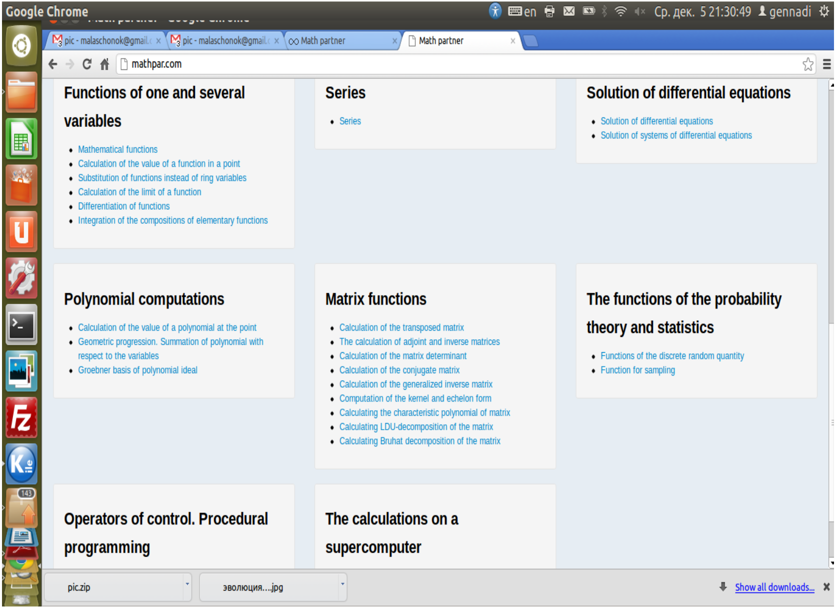
The results: $c = x + 9$; $d = y + 2 * x + 11$.

For each algebra we defined elements $\mathbf{0}$ and $\mathbf{1}$, $-\infty$ and ∞.
For each element a we defined the operation of closure: $a^{\times}$, i.e. the amount of $1 + a + a^2 + a^3 + ...$ For the classical algebras this operation is equivalent to $(1 - a)^{-1}$, for $|a| < 1$.

DEMONSTRATION
of
Math Partner



Google Chrome
en
Ср. дек. 5 21:30:17
gennadi
pic - malaschonok@gmail.c
pic - malaschonok@gmail.c
Math partner
Math partner
mathpar.com
MATH PARTNER
Welcome
Workbook
Help
Handbook
Русский
MATHEMATICAL PARTNER
"Mathpar"
will help you to solve problems using mathematics. You could use it at school, at university and at work. You can save the problem and its solution in the form of text or in the form of image.
mathematical workbook →
Acquaintance and first steps
Input data and run the calculations
Actions with functions
Solution of the algebraic equation
Vectors and matrices
Generation of random elements
Construction of 2D and 3D plots
Plotting functions
Plots 3D of explicit functions
Environment for mathematical objects
Setting of environment
Numerical sets with standard operations
Several numerical sets
Group algebras
Constants
pic.zip
эволюция....jpg
Show all downloads...


Google Chrome
en
Ср. дек. 5 21:30:49
gennadi
pic - malaschonok@gmail.c
pic - malaschonok@gmail.c
Math partner
Math partner
mathpar.com
Functions of one and several variables
Mathematical functions
Calculation of the value of a function in a point
Substitution of functions instead of ring variables
Calculation of the limit of a function
Differentiation of functions
Integration of the compositions of elementary functions
Series
Series
Solution of differential equations
Solution of differential equations
Solution of systems of differential equations
Polynomial computations
Calculation of the value of a polynomial at the point
Geometric progression. Summation of polynomial with respect to the variables
Groebner basis of polynomial ideal
Matrix functions
Calculation of the transposed matrix
The calculation of adjoint and inverse matrices
Calculation of the matrix determinant
Calculation of the conjugate matrix
Calculation of the generalized inverse matrix
Computation of the kernel and echelon form
Calculating the characteristic polynomial of matrix
Calculating LDU-decomposition of the matrix
Calculating Bruhat decomposition of the matrix
The functions of the probability theory and statistics
Functions of the discrete random quantity
Function for sampling
Operators of control. Procedural programming
The calculations on a supercomputer
pic.zip
эволюция....jpg
Show all downloads...

5.5 Differentiation of functions

To differentiate a function $f(x, y, z)$ with lowest variable x, you have to execute one of commands D(f), D(f,x) or D(f,x{ \widehat{ } 1)}. To fine the second deriva
variable y, you have to execute the command D(f,y{ \widehat{ } 2)}. And so on.

To find a mixed first-order derivative of the function f there is a command D(f, [x, y]), to find the derivative of higher order to use the command D(f,[x { \widehat{
{ }\widehat{ } n]}}, where k, m, n indicate the order of the derivative.

```
SPACE = Z[x, y];
f = \sin(x^2 + \tg(y^3 + x));
h = \D(f, y);
\print(h);
```

```
SPACE = Z[x, y];
f = \sin(x^2 + \tg(y^3 + x));
h = \D(f);
\print(h);
```

```
SPACE = Z[x, y, z];
f = x^8y^4z^9;
g = \D(f, [x^2, y^2, z^2]);
\print(g);
```

variable y, you have to execute the command D(f,y{ \widehat{ } 2)}. And so on.

To find a mixed first-order derivative of the function f there is a command D(f, [x, y]), to find the derivative of higher order to use the command D(f,[x { \widehat{ { }\widehat{ } n])}, where k, m, n indicate the order of the derivative.

$$SPACE = Z[x, y];$$
$$f = \mathbf{sin}(x^2 + \mathbf{tg}(y^3 + x));$$
$$h = D_y(f);$$
$$\mathbf{print}(h);$$
$$out:$$
$$h = 3y^2 \cdot \mathbf{cos}(x^2 + \mathbf{tg}(y^3 + x))/(\mathbf{cos}(y^3 + x))^2;$$

$$SPACE = Z[x, y];$$
$$f = \mathbf{sin}(x^2 + \mathbf{tg}(y^3 + x));$$
$$h = D_x(f);$$
$$\mathbf{print}(h);$$
$$out:$$
$$h = (2x \cdot \mathbf{cos}(x^2 + \mathbf{tg}(y^3 + x)) \cdot (\mathbf{cos}(y^3 + x))^2 + \mathbf{cos}(x^2 + \mathbf{tg}(y^3 + x)))/(\mathbf{cos}(y^3 + x))^2;$$

$$SPACE = Z[x, y, z];$$
$$f = x^8 y^4 z^9;$$
$$g = D_{[x^2 y^2 z^2]}(f);$$
$$\mathbf{print}(g);$$
$$out:$$
$$g = 48384 z^7 y^2 x^6;$$

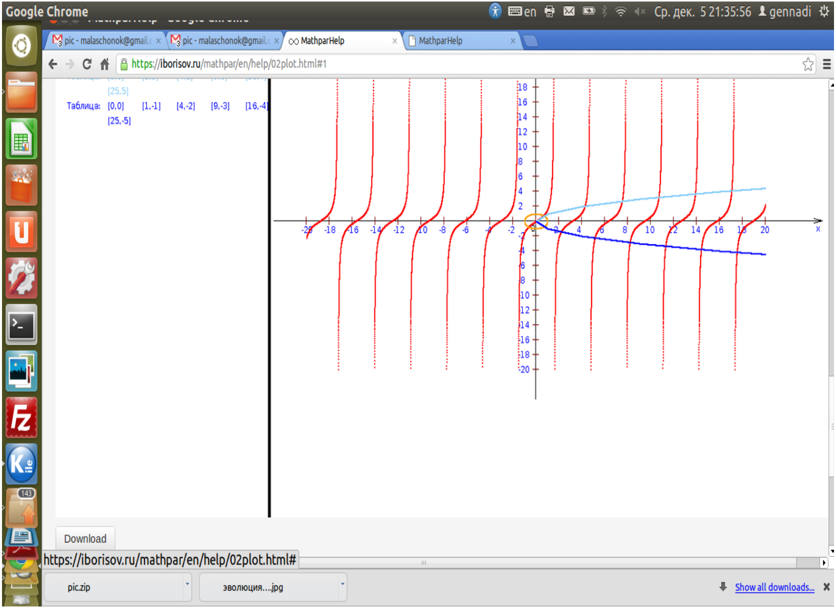
```
    ],
    [-10, 10, -10, 10]);
```

Construction of various plots of functions in one coordinate system

To construct the plots of functions defined in different ways, you must first build a plot of each function and then execute the command showPlots([f_1, f_2, ..., f_n]).

You can specify the signature of the axes of the graph and its caption. It's enough to run showPlots([f1, f2, f3, f4], ['x', 'y', 'title']), instead of specifying 'x' — signature on the axis OX, instead of 'y' — signature on the axis OY, instead of the 'title' \, — the header graphic. Default is ['x' , 'y' , ' '].

```
f1 = \plot(\tg(x), [-20, 20, -20, 20]);
f2=\tablePlot(
    [
      [0, 1, 4, 9, 16, 25],
      [0, 1, 2, 3, 4, 5]
    ],
    [-10, 10, -10, 10]);
f3 = \paramPlot([\sin(x), \cos(x)], [-10, 10]);
f4 = \tablePlot(
    [
      [0, 1, 4, 9, 16, 25],
      [0, -1, -2, -3, -4, -5]
    ],
    [-10, 10, -10, 10]);
\showPlots([f1, f2, f3, f4],
    ['x', 'y', 'The functions f1,  f2,  f3,  f4,  f5']);
```


Google Chrome
Ср. дек. 5 21:35:56
gennadi
pic - malaschonok@gmail.
MathparHelp
MathparHelp
https://iborisov.ru/mathpar/en/help/02plot.html#1
[25,5]
Таблица: [0,0] [1,-1] [4,-2] [9,-3] [16,-4]
[25,-5]
Download
https://iborisov.ru/mathpar/en/help/02plot.html#
pic.zip
эволюция....jpg
Show all downloads...

Google Chrome

Ср. дек. 5 21:36:38 gennadi

pic - malaschonok@gmail.c | pic - malaschonok@gmail.c | MathparHelp | MathparHelp

https://iborisov.ru/mathpar/en/help/02plot.html#1

Download

3.2 Plots 3D of explicit functions

You can build 3D graphs of the functions that are defined explicitly. To obtain the plot 3D of an explicit function $f = f(x, y)$ the command plot3d(f, [x0, x1, y0, y
$[x0, x1]$ is an interval on the axis OX, $[y0, y1]$ is an interval on the axis OY.

The obtained plot can be rotated and to increase or decrease.

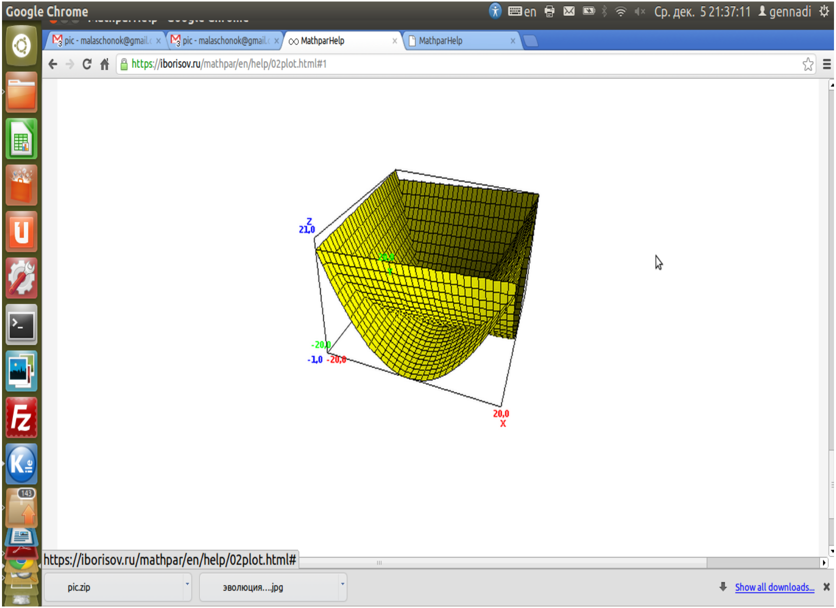
Moving the mouse holding down the left ``mouse" button causes the rotation of the coordinate system of schedule. After stopping the movement of the ``mouse
in the new rotated coordinate system. Moving the mouse holding down the left mouse button while pressing $ *Shift*$ button leads to a change in image scale. A
movement of the ``mouse" graphics are redrawn in the new scale.

```
f = x^2 / 20 + y^2 / 20;
\plot3d(f, [-20, 20, -20, 20]);
```

```
\plot3d([x / 20 + y^2 / 20, x^2 / 20 + y / 20], [-20, 20, -20, 20]);
```

Back to the table of contents

pic.zip | эволюция....jpg | Show all downloads...


Google Chrome
en
Ср. дек. 5 21:37:11
gennadi
pic - malaschonok@gmail.c
pic - malaschonok@gmail.c
MathparHelp
MathparHelp
https://iborisov.ru/mathpar/en/help/02plot.html#1
Z
21,0
-20,0
-1,0
-20,0
20,0
X
143
https://iborisov.ru/mathpar/en/help/02plot.html#
pic.zip
эволюция....jpg
Show all downloads...

7.2 Solution of systems of differential equations

Procedure of solving a system of differential equations (SDE) consists of four parts.

1. To set the ring ($SPACE$).

2. To set a system of equations (systLDE).

3. To set initial conditions (initCond).

4. To get solution of SDE (solveLDE).

```
SPACE=R64[t];
g=\systLDE(\d(x, t)-y+z=0, -x-y+\d(y, t)=0, -x-z+\d(z, t)=0);
f= \initCond(\d(x, t, 0, 0)=1, \d(y, t, 0, 0)=2,
             \d(z, t, 0, 0)=3);
h= \solveLDE(g, f);
\print(h);
```

```
SPACE=R64[t];
g=\systLDE(\d(x, t, 2)+\d(x, t)-\d(y, t)=1,
          \d(x, t)+x-\d(y, t, 2)=1+4\exp(t));
f=\initCond(\d(x, t, 0, 0)=1, \d(x, t, 0, 1)=2,
            \d(y, t, 0, 0)=0, \d(y, t, 0, 1)=1);
h= \solveLDE(g, f);
\print(h);
```

7.2 Solution of systems of differential equations

Procedure of solving a system of differential equations (SDE) consists of four parts.

1. To set the ring ($SPACE$).

2. To set a system of equations (systLDE).

3. To set initial conditions (initCond).

4. To get solution of SDE (solveLDE).

$$SPACE = R64[t];$$

$$g = \begin{cases} x'_t - y + z & = & 0 \\ -x - y + y'_t & = & 0 \\ -x - z + z'_t & = & 0 \end{cases};$$

$$f = \begin{cases} x^{(0)}_{t=0} & = & 1 \\ y^{(0)}_{t=0} & = & 2 \\ z^{(0)}_{t=0} & = & 3 \end{cases};$$

$$h = \mathbf{solveLDE}(g, f);$$

$$\mathbf{print}(h);$$

$out:$

$$h = [((-2) + 5.00 \cdot e^t + (-1.00 \cdot e^t) \cdot t), (2 + (-1.00 \cdot e^t)), ((-2) + 4.00 \cdot e^t + (-1.00 \cdot e^t) \cdot t)];$$

Google Chrome

pic - malaschonok@gmail. | pic - malaschonok@gmail. | MathparHelp

https://iborisov.ru/mathpar/en/help/06dequation.html#1

```
SPACE=R64[t];
g=\systLDE(\d(x, t)-8y+x=0, \d(y, t)-x-y=0);
f= \initCond(\d(x, t, 0, 0)=a, \d(y, t, 0, 0)=b);
h= \solveLDE(g, f);
\print(h);
```

```
SPACE=R64[t];
g=\systLDE(\d(x, t)+3x-4y=9(\exp(t))^2, \d(y, t)+2x-3y=3(\exp(t))^2);
f= \initCond(\d(x, t, 0, 0)=2, \d(y, t, 0, 0)=0);
h= \solveLDE(g, f);
\print(h);
```

```
SPACE=R64[t];
g=\systLDE(\d(x, t, 2)+\d(y, t)=\sh(t)-\sin(t)-t, \d(y, t, 2)-\d(x, t)=\ch(t)-\cos(t));
f=\initCond(\d(x, t, 0, 0)=2, \d(x, t, 0, 1)=0,
            \d(y, t, 0, 0)=0, \d(y, t, 0, 1)=1);
h=\solveLDE(g, f);
\print(h);
```

```
SPACE=R64[t];
g=\systLDE(\d(x, t)+5y-4x=0, \d(y, t)-x=0);
f= \initCond(\d(x, t, 0, 0)=0, \d(y, t, 0, 0)=1);
```

pic.zip | эволюция....jpg | Show all downloads...

Google Chrome   en   Ср. дек. 5 21:39:58 gennadi

pic - malaschonok@gmail.c | pic - malaschonok@gmail.c | MathparHelp

https://iborisov.ru/mathpar/en/help/06dequation.html#1

$$SPACE = R64[t];$$

$$g = \begin{cases} x'_t - 8y + x & = 0 \\ y'_t - x - y & = 0 \end{cases};$$

$$f = \begin{cases} x^{(0)}_{t=0} & = a \\ y^{(0)}_{t=0} & = b \end{cases};$$

$$h = \mathbf{solveLDE}(g, f);$$

$$\mathbf{print}(h);$$

out :

$$h = [(((-48.00 \cdot b) + 24.00 \cdot a)/36.00 \cdot e^{-3.00t} + (48.00 \cdot b + 12.00 \cdot a)/36.00 \cdot e^{3.00t}), (((-6.00 \cdot a) + 12.00 \cdot b)/36.00 \cdot e^{-3.00t} + (6.00 \cdot a + 24.($$

$$SPACE = R64[t];$$

$$g = \begin{cases} x'_t + 3x - 4y & = 9(e^t)^2 \\ y'_t + 2x - 3y & = 3(e^t)^2 \end{cases};$$

$$f = \begin{cases} x^{(0)}_{t=0} & = 2 \\ y^{(0)}_{t=0} & = 0 \end{cases};$$

$$h = \mathbf{solveLDE}(g, f);$$

$$\mathbf{print}(h);$$

out :

$$h = [(e^t + e^{2.00t}), (e^t + (-1.00 \cdot e^{2.00t}))];$$

```
SPACE=R64[t];
```

pic.zip   эволюция....jpg   Show all downloads...

9.8 Calculating LDU-decomposition of the matrix

To calculate the LDU-decomposition of the matrix A, you must run LDU(A).

The result is a vector of three matrices $[L, D, U]$. Where L is a lower triangular matrix, U — upper triangular matrix, D — permutation matrix, multiplied by the matrix. If the elements of the matrix A are elements of commutative domain R, then elements of matrices L, D^{-1}, U are elements of the same domain R.

```
SPACE=Z[x];
A=[[0, 1, 0], [4, 5, 1],[1, 1, 1]];
B=\LDU(A);
\print(B);
```

```
SPACE=Z[x];
A=[[1, 4,0,1], [4, 5,5,3],[1,2,2,2],[3,0,0,1]];
B=\LDU(A);
\print(B);
```

```
SPACE=Z[x,y];
A=[[\cos(y),\sin(x)],[\sin(y),\cos(x)]];
B=\LDU(A);
\print(B);
```

Google Chrome

pic - malaschonok@gmail. | pic - malaschonok@gmail. | MathparHelp

https://iborisov.ru/mathpar/en/help/08matrix.html#7

9.8 Calculating LDU-decomposition of the matrix

To calculate the LDU-decomposition of the matrix A, you must run LDU(A).

The result is a vector of three matrices $[L, D, U]$. Where L is a lower triangular matrix, U — upper triangular matrix, D — permutation matrix, multiplied by the matrix. If the elements of the matrix A are elements of commutative domain R, then elements of matrices L, D^{-1}, U are elements of the same domain R.

$$SPACE = Z[x];$$

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 4 & 5 & 1 \\ 1 & 1 & 1 \end{pmatrix};$$

$$B = \mathbf{LDU}(A);$$

$$\mathbf{print}(B);$$

out :

$$B = \left[\begin{pmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ -1 & 1 & 3 \end{pmatrix}, \begin{pmatrix} 0 & (1/16) & 0 \\ (1/4) & 0 & 0 \\ 0 & 0 & (1/12) \end{pmatrix}, \begin{pmatrix} 4 & 5 & 1 \\ 0 & 4 & 0 \\ 0 & 0 & 3 \end{pmatrix}\right];$$

https://iborisov.ru/mathpar/en/help/08matrix.html#

pic.zip | эволюция....jpg | Show all downloads...

```
A=[[\cos(y),\sin(x)],[\sin(y),\cos(x)]];
B=\LDU(A);
\print(B);
```

9.9 Calculating Bruhat decomposition of the matrix

To calculate the Bruhat decomposition of the matrix A, you must run BruhatDecomposition(A).

The result is a vector of three matrices $[V, D, U]$. Where V and U — upper triangular matrices, D — permutation matrix, multiplied by the inverse of the diago elements of the matrix A are elements of commutative domain R, then elements of matrices V, D^{-1}, U are elements of the same domain R.

```
SPACE=Z[x];
A=[[1, 4,0,1], [4, 5,5,3],[1,2,2,2],[3,0,0,1]];
B=\BruhatDecomposition(A);
\print(B);
```

```
SPACE=Z[x,y];
A=[[\cos(y),\sin(x)],[\sin(y),\cos(x)]];
B=\BruhatDecomposition(A);
\print(B);
```

Back to the table of contents

Google Chrome Ср. дек. 5 21:41:54 gennadi

pic - malaschonok@gmail.c | pic - malaschonok@gmail.c | MathparHelp

https://iborisov.ru/mathpar/en/help/08matrix.html#7

The result is a vector of three matrices $[V, D, U]$. Where V and U — upper triangular matrices, D — permutation matrix, multiplied by the inverse of the diago
elements of the matrix A are elements of commutative domain R, then elements of matrices V, D^{-1}, U are elements of the same domain R.

$$SPACE = Z[x];$$

$$A = \begin{pmatrix} 1 & 4 & 0 & 1 \\ 4 & 5 & 5 & 3 \\ 1 & 2 & 2 & 2 \\ 3 & 0 & 0 & 1 \end{pmatrix};$$

$$B = \mathbf{BruhatDecomposition}(A);$$

$$\mathbf{print}(B);$$

out :

$$B = \left[\begin{pmatrix} -24 & 0 & 12 & 1 \\ 0 & 60 & 15 & 4 \\ 0 & 0 & 6 & 1 \\ 0 & 0 & 0 & 3 \end{pmatrix}, \begin{pmatrix} 0 & 0 & (1/-144) & 0 \\ 0 & 0 & 0 & (1/-1440) \\ 0 & (1/18) & 0 & 0 \\ (1/3) & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 3 & 0 & 0 & 1 \\ 0 & 6 & 6 & 5 \\ 0 & 0 & -24 & -16 \\ 0 & 0 & 0 & 60 \end{pmatrix}\right];$$

```
SPACE=Z[x,y];
```

pic.zip | эволюция....jpg | Show all downloads...