

1. Стандартные процедуры.

```
> restart; with(LinearAlgebra): mylib := table():
```

1.1. Вспомогательные процедуры (используются только в стандартных).

pprintf

Функция, аналогичная по синтаксису printf, но выводящая алгебраические выражения в формате prettyprint (используется только в стандартных процедурах).

```
> mylib[pprintf] := proc(S::string)
    uses StringTools, Typesetting; local X,L,r,n,i,j,k;
    L := [RegSplit("%a",S)]; X := [_passed[2..-1]]; r := NULL; j := 1;
    for i from 1 to nops(L) do
        n := CountCharacterOccurrences(L[i],"%");
        r := r, mi(sprintf(L[i],seq(X[k],k=j..j+n-1))); j := j+n;
        if j <= nops(X) then r := r, op(map(Typeset,[X[j]])); j := j+1; end if; end do;
    print(mrow(r)); end proc;
```

P

Кубический многочлен с коэффициентами a,b,c,d от переменных x,y.

```
> mylib[P] := (a,b,c,d,x,y) -> a*x^3+b*x^2*y+c*x*y^2+d*y^3:
```

delta

Определитель матрицы $\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix}$.

```
> mylib[delta] := (x1,y1,x2,y2) -> x1*y2-x2*y1:
```

R

Результат системы с матрицей $\begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \end{bmatrix}$.

```
> mylib[R] := proc(a1,b1,c1,d1,a2,b2,c2,d2)
    local dab,dac,dad,dbc,dbd,dcd,Rez;
    dab := factor(a1*b2-a2*b1); dac := factor(a1*c2-a2*c1);
    dad := factor(a1*d2-a2*d1); dbc := factor(b1*c2-b2*c1);
    dbd := factor(b1*d2-b2*d1); dcd := factor(c1*d2-c2*d1);
    Rez := factor(dad^3-2*dab*dad*dcd+dac^2*dcd+dab*dbd^2-dab*dbc*dcd-dac*dad*dbd);
    mylib[pprintf] ("Resultant: %a", Rez);
end proc;
```

1.2. Процедуры для однородных кубических систем с примерами их действия.

zamproc

Производит линейную замену $\begin{bmatrix} r_1 & s_1 \\ r_2 & s_2 \end{bmatrix}$ в системе с матрицей $\begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \end{bmatrix}$.

Результат --- матрица 2x4 получаемой системы.

При написании full = true будет дополнительно выписана матрица исходной системы, замена и ее определитель.

При написании res = true будет дополнительно выписан результат исходной системы.

При написании lbl = true элементы матрицы получаемой системы будут выписаны построчно.

```
> mylib[zamproc] := proc(a1,b1,c1,d1,a2,b2,c2,d2, r1,s1,r2,s2, {full::boolean := false,
    res::boolean := false, lbl::boolean := false}) :: Matrix;
    local alnn,b1nn,c1nn,d1nn,a2nn,b2nn,c2nn,d2nn,aln,b1n,c1n,d1n,a2n,b2n,c2n,d2n,Delta,
    i1,i2,i3,i4,i5,i6,i7,i8,ix,m1,m2,m3,m4,m5,m6,m7,m8,mx,M; Delta := r1*s2 - s1*r2;
    aln := (s2*mylib[P] (a1,b1,c1,d1,r1,r2)-s1*mylib[P] (a2,b2,c2,d2,r1,r2))/Delta;
    b1n := (3*a1*r1^2*s1*s2+2*b1*r1*s1*r2*s2+b1*r1^2*s2^2+2*c1*r1*r2*s2^2+c1*s1*r2^2*s2
    +3*d1*r2^2*s2^2-3*a2*r1^2*s1^2-2*b2*r1*s1^2*r2-b2*r1^2*s1*s2-2*c2*r1*s1*r2*s2
    -c2*s1^2*r2^2-3*d2*s1*r2^2*s2)/Delta;
    c1n := (3*a1*r1*s1^2*s2+2*b1*r1*s1*s2^2+b1*s1^2*r2*s2+2*c1*s1*r2*s2^2+c1*r1*s2^3
    +3*d1*r2*s2^3-3*a2*r1*s1^3-2*b2*r1*s1^2*s2-b2*s1^3*r2-2*c2*s1^2*r2*s2
    -c2*r1*s1*s2^2-3*d2*s1*r2*s2^2)/Delta;
    d1n := (s2*mylib[P] (a1,b1,c1,d1,s1,s2)-s1*mylib[P] (a2,b2,c2,d2,s1,s2))/Delta;
    a2n := (-r2*mylib[P] (a1,b1,c1,d1,r1,r2)+r1*mylib[P] (a2,b2,c2,d2,r1,r2))/Delta;
    b2n := (3*a2*r1^3*s1+2*b2*r1^2*s1*r2+b2*r1^3*s2+2*c2*r1^2*r2*s2+c2*r1*s1*r2^2
    +3*d2*r1*r2^2*s2-3*a1*r1^2*s1*r2-2*b1*r1*s1*r2^2-b1*r1^2*r2*s2-2*c1*r1*r2^2*s2
    -c1*s1*r2^3-3*d1*r2^3*s2)/Delta;
    c2n := (3*a2*r1^2*s1^2+2*b2*r1^2*s1*s2+b2*r1*s1^2*r2+2*c2*r1*s1*r2*s2+c2*r1^2*s2^2
    +3*d2*r1*r2*s2^2-3*a1*r1*s1^2*r2-2*b1*r1*s1*r2*s2-b1*s1^2*r2^2-2*c1*s1*r2^2*s2
    -c1*r1*r2*s2^2-3*d1*r2^2*s2^2)/Delta;
    d2n := (-r2*mylib[P] (a1,b1,c1,d1,s1,s2)+r1*mylib[P] (a2,b2,c2,d2,s1,s2))/Delta;
    alnn := simplify(factor(simplify(normal(simplify(aln))))):
```

```

b1nn := simplify(factor(simplify(normal(simplify(b1n))))):
c1nn := simplify(factor(simplify(normal(simplify(c1n))))):
d1nn := simplify(factor(simplify(normal(simplify(d1n))))):
a2nn := simplify(factor(simplify(normal(simplify(a2n))))):
b2nn := simplify(factor(simplify(normal(simplify(b2n))))):
c2nn := simplify(factor(simplify(normal(simplify(c2n))))):
d2nn := simplify(factor(simplify(normal(simplify(d2n))))):
a1n := simplify(factor(simplify(normal(simplify(a1))))):
b1n := simplify(factor(simplify(normal(simplify(b1))))):
c1n := simplify(factor(simplify(normal(simplify(c1))))):
d1n := simplify(factor(simplify(normal(simplify(d1))))):
a2n := simplify(factor(simplify(normal(simplify(a2))))):
b2n := simplify(factor(simplify(normal(simplify(b2))))):
c2n := simplify(factor(simplify(normal(simplify(c2))))):
d2n := simplify(factor(simplify(normal(simplify(d2))))):
M := Matrix(2, 4); M(1,1) := a1nn; M(1,2) := b1nn; M(1,3) := c1nn; M(1,4) := d1nn;
      M(2,1) := a2nn; M(2,2) := b2nn; M(2,3) := c2nn; M(2,4) := d2nn;
if full then mylib[pprintf] ("Initial system:"); print(a1n,b1n,c1n,d1n); print(a2n,b2n,c2n,d2n); fi;
if res then mylib[R](a1, b1, c1, d1, a2, b2, c2, d2); end if;
if full then mylib[pprintf] ("substitution: %a, %a; %a, %a",
      simplify(factor(simplify(r1))),simplify(factor(simplify(s1))),
      simplify(factor(simplify(r2))),simplify(factor(simplify(s2))));
      mylib[pprintf] ("det: %a", simplify(factor(simplify(mylib[delta](r1,s1,r2,s2))));
      mylib[pprintf] ("system after substitution:"); end if;
if lbl then print(a1nn); print(b1nn); print(c1nn); print(d1nn);
      print(a2nn); print(b2nn); print(c2nn); print(d2nn);
else print(a1nn,b1nn,c1nn,d1nn); print(a2nn,b2nn,c2nn,d2nn); end if;
return M; end proc;

```

```
> mylib[zamproc](u,0,0,v,0,1,0,1, r1,1,0,s2):
```

$$\frac{u r l^2, 3 r l u - r l, 3 u - 2, \frac{v s^2^3 - s^2^2 + u - 1}{r l}}{0, r l^2, 2 r l, s^2^2 + 1}$$

Вызов процедуры с включением всех дополнительных опций:

вывод исходной системы и замены с ее определителем (full = true), вывод результата исходной системы (res = true), построчный вывод элементов матрицы получаемой системы (lbl = true).

```
> mylib[zamproc](u,0,0,v,0,1,0,1, r1,1,0,s2, full = true, res = true, lbl = true):
```

Initial system:
 $u, 0, 0, v$
 $0, 1, 0, 1$
Resultant: $u (u^2 + v^2)$
substitution: $r l, 1; 0, s 2$
det: $r l s 2$
system after substitution:
 $u r l^2$
 $3 r l u - r l$
 $3 u - 2$
 $\frac{v s^2^3 - s^2^2 + u - 1}{r l}$
 0
 $r l^2$
 $2 r l$
 $s^2^2 + 1$

zamproc11

Замена $\begin{bmatrix} r_1 & s_1 \\ r_2 & s_2 \end{bmatrix}$ в системе с $l = 1$, записанной в виде (α, β) , $\begin{bmatrix} p_1 & q_1 & t_1 \\ p_2 & q_2 & t_2 \end{bmatrix}$.

Результат -- матрица 2x4 полученной системы.

При написании full = true будет дополнительно выписана матрица исходной системы, замена и ее определитель.

При написании res = true будет дополнительно выписан результат исходной матрицы 2x3.

При написании lbl = true элементы матрицы получаемой системы будут выписаны построчно.

```
> mylib[zamproc11] := proc(alp,beta,p1,q1,t1,p2,q2,t2,r1,s1,r2,s2, {full::boolean := false,
      res::boolean := false, lbl::boolean := false}) :: Matrix;
```

```

local a1n,b1n,c1n,d1n,a2n,b2n,c2n,d2n,result,p1n,q1n,t1n,p2n,q2n,t2n,res2,M,G;
a1n := factor(alp*p1); b1n := factor(alp*q1+beta*p1); c1n := factor(alp*t1+beta*q1);
d1n := factor(beta*t1); a2n := factor(alp*p2); b2n := factor(alp*q2+beta*p2);
c2n := factor(alp*t2+beta*q2); d2n := factor(beta*t2);
p1n := factor((r1^2*s2*p1-r1^2*s1*p2+r2*q1*r1*s2-r1*q2*s1*r2+r2^2*s2*t1-r2^2*s1*t2)/(r1*s2-r2*s1));
q1n := factor((2*r1*s1*s2*p1-2*s1^2*r1*p2+s2^2*q1*r1+s1*s2*r2*q1-s1*s2*r1*q2-s1^2*q2*r2+2*s2^2*r2*t1-2*r2*s2*s1*t2)/(r1*s2-r2*s1));
t1n := factor((s1^2*s2*p1-s1^3*p2+s1*s2^2*q1-s1^2*s2*q2+s2^3*t1-s2^2*s1*t2)/(r1*s2-s1*r2));
p2n := factor(-(r1^2*r2*p1-r1^3*p2+r1*r2^2*q1-r1^2*r2*q2+r2^3*t1-r2^2*r1*t2)/(r1*s2-r2*s1));
q2n := factor(-(2*r1*s1*r2*p1-2*r1^2*s1*p2+r2*q1*r1*s2+r2^2*q1*s1-r1^2*q2*s2-r1*q2*s1*r2+2*r2^2*s2*t1-2*r2*s2*r1*t2)/(r1*s2-r2*s1));
t2n := factor(-(s1^2*r2*p1-s1^2*r1*p2+s1*s2*r2*q1-s1*s2*r1*q2+s2^2*r2*t1-s2^2*r1*t2)/(r1*s2-r2*s1));
res2 := factor((p1n*t2n-p2n*t1n)^2-(p1n*q2n-p2n*q1n)*(q1n*t2n-q2n*t1n));
if res then G := Matrix([[factor(p1),factor(q1),factor(t1)], [factor(p2),factor(q2),factor(t2)]]);
  result := factor((p1*t2-p2*t1)^2-(p1*q2-p2*q1)*(q1*t2-q2*t1)); mylib[pprintf]("G = %a", G);
  mylib[pprintf]("Resultant of G: %a", result); end if;
M := mylib[zamproc](a1n,b1n,c1n,d1n,a2n,b2n,c2n,d2n,r1,s1,r2,s2,
  ':-full'=full,':-res'=false,':-lbl'=lbl); return M; end proc:

```

```

> mylib[zamprocl1](1,-1,2,1,3,0,1,1, r1,0,r2,s2):

$$\frac{r1}{r1} \frac{(r1-r2)(2r1^2+r1r2+3r2^2)}{r1s2}, -\frac{s2}{r1} \frac{(r1^2-4r1r2+9r2^2)}{r1}, \frac{s2^2}{r1} \frac{(2r1-9r2)}{r1}, -\frac{3s2^3}{r1}$$


$$-\frac{r2}{r1} \frac{(r1-r2)(r1^2+3r2^2)}{r1s2}, \frac{r1^3+r1^2r2-7r1r2^2+9r2^3}{r1}, -\frac{s2}{r1} \frac{r2(5r1-9r2)}{r1}, -\frac{s2^2}{r1} \frac{(r1-3r2)}{r1}$$


```

11pG

Замена $\begin{bmatrix} r_1 & s_1 \\ r_2 & s_2 \end{bmatrix}$ в системе с $l=1$, записанной в виде (α, β) , $\begin{bmatrix} p_1 & q_1 & t_1 \\ p_2 & q_2 & t_2 \end{bmatrix}$.

Результат --- вектор коэффициентов общего множителя p_0^1 и матрица G получаемой системы.

При написании full = true будут дополнительно выписаны p_0^1 и G исходной системы, замена и ее определитель.

```

> mylib[11pG] := proc(alp,beta, p1,q1,t1,p2,q2,t2,r1,s1,r2,s2, {full::boolean := false})
local alpn,betan,p1n,q1n,t1n,p2n,q2n,t2n,p10,G;
alpn := factor(alp*r1+beta*r2); betan := factor(alp*s1+beta*s2);
p1n := factor((r1^2*s2*p1-r1^2*s1*p2+r2*q1*r1*s2-r1*q2*s1*r2+r2^2*s2*t1-r2^2*s1*t2)/(r1*s2-r2*s1));
q1n := factor((2*r1*s1*s2*p1-2*s1^2*r1*p2+s2^2*q1*r1+s1*s2*r2*q1-s1*s2*r1*q2-s1^2*q2*r2+2*s2^2*r2*t1-2*r2*s2*s1*t2)/(r1*s2-r2*s1));
t1n := factor((s1^2*s2*p1-s1^3*p2+s1*s2^2*q1-s1^2*s2*q2+s2^3*t1-s2^2*s1*t2)/(r1*s2-s1*r2));
p2n := factor(-(r1^2*r2*p1-r1^3*p2+r1*r2^2*q1-r1^2*r2*q2+r2^3*t1-r2^2*r1*t2)/(r1*s2-r2*s1));
q2n := factor(-(2*r1*s1*r2*p1-2*r1^2*s1*p2+r2*q1*r1*s2+r2^2*q1*s1-r1^2*q2*s2-r1*q2*s1*r2+2*r2^2*s2*t1-2*r2*s2*r1*t2)/(r1*s2-r2*s1));
t2n := factor(-(s1^2*r2*p1-s1^2*r1*p2+s1*s2*r2*q1-s1*s2*r1*q2+s2^2*r2*t1-s2^2*r1*t2)/(r1*s2-r2*s1));
p10 := convert(Matrix([[alp,beta]]),list);
G := Matrix([[factor(p1),factor(q1),factor(t1)], [factor(p2),factor(q2),factor(t2)]]);
if full then mylib[pprintf]("Initial coefficients: %a = %a", (p[0])^(1.), p10);
  mylib[pprintf]("G = %a", G); mylib[pprintf]("substitution: %a, %a; %a, %a", r1, s1, r2, s2);
  mylib[pprintf]("det: %a", simplify(factor(simplify(delta(r1,s1,r2,s2))));
  mylib[pprintf]("coefficients after substitution:"); end if;
p10 := convert(Matrix([[alpn,betan]]),list);
G := Matrix([[factor(p1n),factor(q1n),factor(t1n)], [factor(p2n),factor(q2n),factor(t2n)]]);
mylib[pprintf]("a = %a", (p[0])^(1.), p10); mylib[pprintf]("G = %a", G); end proc:

```

```

> mylib[11pG](1,-1,2,1,3,0,1,1, r1,0,r2,s2):

$$p_0^1 = [r1 - r2, -s2]$$


```

$$G = \begin{bmatrix} \frac{2r1^2+r1r2+3r2^2}{r1} & \frac{s2(r1+6r2)}{r1} & \frac{3s2^2}{r1} \\ -\frac{r2(r1^2+3r2^2)}{r1s2} & \frac{(r1+3r2)(r1-2r2)}{r1} & \frac{s2(r1-3r2)}{r1} \end{bmatrix}$$

l1syst

Представление системы с $l = 1$, записанной в виде (α, β) , $\begin{bmatrix} p_1 & q_1 & t_1 \\ p_2 & q_2 & t_2 \end{bmatrix}$, через матрицу 2×4 .

При написании `lbl = true` элементы матрицы 2×4 будут выписаны построчно.

```
> mylib[l1syst] := proc(alp,beta,p1,q1,t1,p2,q2,t2, {res::boolean := false, lbl::boolean := false})
  local aln,b1n,c1n,d1n,a2n,b2n,c2n,d2n,result,i1,i2,i3,i4,i5,i6,i7,i8,ix,p10,G;
  aln := alp*p1: b1n := alp*q1+beta*p1: c1n := alp*t1+beta*q1: d1n := beta*t1:
  a2n := alp*p2: b2n := alp*q2+beta*p2: c2n := alp*t2+beta*q2: d2n := beta*t2:
  p10 := convert(Matrix([[alp,beta]]),list);
  G := Matrix([[factor(p1),factor(q1),factor(t1)], [factor(p2),factor(q2),factor(t2)]]);
  mylib[pprintf]("Initial coefficients: %a = %a", (p[0])^(1.), p10); mylib[pprintf]("G = %a", G);
  if res then result := factor((p1*t2-p2*t1)^2-(p1*q2-p2*q1)*(q1*t2-q2*t1));
    mylib[pprintf]("Resultant of G: %a", result); end if;
  mylib[pprintf]("Matrix (2x4):");
  if lbl then print(aln); print(b1n); print(c1n); print(d1n);
    print(a2n); print(b2n); print(c2n); print(d2n);
  else print(aln,b1n,c1n,d1n); print(a2n,b2n,c2n,d2n); end if; end proc;
```

```
> mylib[l1syst](1,-1,2,1,3,0,1,1):
```

Initial coefficients: $p_0^1 = [1, -1]$

$$G = \begin{bmatrix} 2 & 1 & 3 \\ 0 & 1 & 1 \end{bmatrix}$$

Matrix (2x4):

2, -1, 2, -3
0, 1, 0, -1

pm1

В системе с $l = 1$ с матрицей $\begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \end{bmatrix}$ выделяются стока p_0^1 и матрица G с ее результатом.

По умолчанию выводится только результат G . При написании `full = true` выписываются все указанные параметры.

```
> mylib[pm1] := proc(a1,b1,c1,d1,a2,b2,c2,d2, {full::boolean := false})
  local P1n,P2n,rez2n,chis,znam,obw,om,gp,mn;
  P1n := factor(simplify(mylib[P](a1,b1,c1,d1,x1,x2)));
  P2n := factor(simplify(mylib[P](a2,b2,c2,d2,x1,x2)));
  chis := numer(P1n/P2n); znam := denom(P1n/P2n); obw := simplify(P1n/chis);
  if coeff(obw,x1)=1 then om := Matrix([[coeff(obw, x1), coeff(obw, x2)]]);
    gp := Matrix([[factor(simplify(coeff(chis, x1^2))),factor(simplify(coeff(chis, x1)/x2)),
      factor(simplify(coeff(chis, x2^2))),[factor(simplify(coeff(znam, x1^2))),
      factor(simplify(coeff(znam, x1)/x2)),factor(simplify(coeff(znam, x2^2)))]]);
  else mn := coeff(obw,x1); om := Matrix([[coeff(obw, x1)/mn, coeff(obw, x2)/mn]]);
    gp := Matrix([[factor(simplify(coeff(chis, x1^2))*mn),factor(simplify(coeff(chis, x1)/x2)*mn),
      factor(simplify(coeff(chis, x2^2))*mn)], [factor(simplify(coeff(znam, x1^2))*mn),
      factor(simplify(coeff(znam, x1)/x2)*mn),factor(simplify(coeff(znam, x2^2))*mn)]]);
  if om[1,1]<0 then om := -om: gp := -gp fi: fi:
  rez2n := factor(simplify(combine((gp[1, 1]*gp[2, 3]-gp[1, 3]*gp[2, 1])^2-(gp[1, 1]*gp[2, 2]-gp[1, 2]*
  gp[2, 1])*(gp[1, 2]*gp[2, 3]-gp[1, 3]*gp[2, 2])))); om := convert(om,list);
  if full=true then mylib[pprintf]("Initial coefficients:");
    print(a1, b1, c1, d1); print(a2, b2, c2, d2); mylib[pprintf]("Factorization:");
    mylib[pprintf]("a = %a", (p[0])^(1.), om); mylib[pprintf]("G = %a", gp); end if:
  mylib[pprintf]("Resultant of G: %a", rez2n); end proc;
```

```
> mylib[pm1](2,1,3,0,0,1,1,0, full=true);
```

Initial coefficients:

2, 1, 3, 0
0, 1, 1, 0

Factorization:

$p_0^1 = [1, 0]$

$$G = \begin{bmatrix} 2 & 1 & 3 \\ 0 & 1 & 1 \end{bmatrix}$$

Resultant of G: 8

```
> save(mylib, "newlib.m");
```

При компиляции создается файл `newlib.m`, из которого в дальнейшем подключается список `mylib`.