Axel Kemper

# Verfahren von Makarov zur Multiplikation von 5x5 Matrizen

O.M. Makarov hat 1985 ein Verfahren veröffentlicht, dass zwei 5x5 Matrizen miteinander multipliziert und dabei 100 Elementarmultiplikationen benötigt. Dieses Papier beschreibt das Verfahren.

Es soll gelten:

$$B * A = L$$

Achtung! Nicht A*B sondern B*A!

mit

$$B = \begin{Vmatrix} b1 & b2 & b3 & b4 & b5 \\ c1 & c2 & c3 & c4 & c5 \\ g1 & g2 & g3 & g4 & g5 \\ d1 & d2 & d3 & d4 & d5 \\ r1 & r2 & r3 & r4 & r5 \end{Vmatrix}$$

$$A = \begin{Vmatrix} a11 & a12 & a13 & a14 & a15 \\ a21 & a22 & a23 & a24 & a25 \\ a31 & a32 & a33 & a34 & a35 \\ a41 & a42 & a43 & a44 & a45 \\ a51 & a52 & a53 & a54 & a55 \end{Vmatrix}$$

$$L = \begin{Vmatrix} k1 & k2 & k3 & k4 & k5 \\ m1 & m2 & m3 & m4 & m5 \\ p1 & p2 & p3 & p4 & p5 \\ n1 & n1 & n3 & n4 & n5 \\ l1 & l2 & l3 & l4 & l5 \end{Vmatrix}$$

Es soll gelten:

$$L_1 = L_2 + L_3 + L_4$$

mit

$$L_1 = \begin{Vmatrix} k1 & k2 & k3 & k4 & k5 \\ m1 - m2 & m2 & m3 - m4 & m4 & m5 \\ p1 & p2 & p3 & p4 & p5 \\ n1 - n2 & n2 & n3 - n4 & n4 & n5 \\ l1 & l2 & l3 & l4 & l5 \end{Vmatrix}$$

$$L_2 = \begin{Vmatrix} 0 & k21 & 0 & k41 & 0 \\ 0 & m21 & 0 & m41 & 0 \\ 0 & p21 & 0 & p41 & 0 \\ 0 & n21 & 0 & n41 & 0 \\ 0 & l21 & 0 & l41 & 0 \end{Vmatrix}$$

Stand: 6. Januar 2014

$$L_3 = \left\| \begin{matrix} k12 & k22 & k32 & k42 & k52 \\ m12-k22 & k22 & m32-k42 & k42 & m52 \\ p12 & p22 & p32 & p42 & p52 \\ n12-p22 & p22 & n32-p42 & p42 & n52 \\ l12 & 0 & l32 & 0 & l52 \end{matrix} \right\|$$

$$L_4 = \left\| \begin{matrix} k13 & k23 & k33 & k43 & k53 \\ k13-k23 & k23 & k33-k43 & k43 & m53 \\ p13 & p23 & p33 & p43 & p53 \\ p13-p23 & p23 & p33-p43 & p43 & n53 \\ l13 & 0 & l33 & 0 & l53 \end{matrix} \right\|$$

Aus Matrix $L_1$ kann man die Ergebnismatrix L durch Umformung ermitteln.

$$\left\| \begin{matrix} b1+b2-c1-c2 & b3+b4-c3-c4 & b5 \\ g1+g2-d1-d2 & g3+g4-d3-d4 & g5 \\ r1+r2 & r3+r4 & r5 \end{matrix} \right\| * \left\| \begin{matrix} a12 & a14 \\ a32 & a34 \\ a52 & a54 \end{matrix} \right\| = \left\| \begin{matrix} k21 & k41 \\ p21 & p41 \\ l21^1 & l41^1 \end{matrix} \right\| \quad (1)$$

$$\left\| \begin{matrix} b2-c2 & b4-c4 & c5 \\ g2-d2 & g4-d4 & d5 \\ -r2 & -r4 & 0 \end{matrix} \right\| * \left\| \begin{matrix} a12-a22 & a14-a24 \\ a32-a42 & a34-a44 \\ a52 & a54 \end{matrix} \right\| = \left\| \begin{matrix} m21 & m41 \\ n21 & n41 \\ l21^2 & l41^2 \end{matrix} \right\| \quad (2)$$

$$l21 = l21^1 + l21^2, \; l41 = l41^1 + l41^2 \quad (3)$$

Die Beziehungen (1) – (3) ermitteln die Matrix $L_2$

$$\left\| \begin{matrix} b1 & b3 & b5 \\ g1 & g3 & g5 \\ r1 & r3 & r5 \end{matrix} \right\| * \left\| \begin{matrix} a11 & a13 & a15 \\ a31 & a33 & a35 \\ a51 & a53 & a55 \end{matrix} \right\| = \left\| \begin{matrix} k12 & k32 & k52 \\ p12 & p32 & p52 \\ l12 & l32 & l52 \end{matrix} \right\| \quad (4)$$

$$\left\| \begin{matrix} c1 & c3 & c5 \\ d1 & d3 & d5 \end{matrix} \right\| * \left\| \begin{matrix} v1 & v2 & a15-a25 \\ v3 & v4 & a35-a45 \\ a51-a52 & a53-a54 & a55 \end{matrix} \right\| = \left\| \begin{matrix} m12-k22 & m32-k42 & m52 \\ n12-p22 & n32-p42 & n52 \end{matrix} \right\|$$

(5)

$$\left\| \begin{matrix} b2-c1-c2 & b4-c3-c4 \\ g2-d1-d2 & g4-d3-d4 \end{matrix} \right\| * \left\| \begin{matrix} a22-a12-a21 & a24-a23-a14 \\ a42-a41-a32 & a44-a43-a34 \end{matrix} \right\| = \left\| \begin{matrix} k22 & k42 \\ p22 & p42 \end{matrix} \right\| \quad (6)$$

$$v1 = a11 - a21 - a12 + a22, \; v2 = a13 - a23 - a14 + a24, \quad (7a)$$

$$v3 = a31 - a41 - a32 + a42, \; v4 = a33 - a43 - a34 + a44 \quad (7b)$$

Die Beziehungen (4) – (7b) ermitteln die Matrix $L_3$

$$\left\| \begin{matrix} b2 & b4 \\ g2 & g4 \\ r2 & r4 \end{matrix} \right\| * \left\| \begin{matrix} a21 & a23 & a25 \\ a41 & a43 & a45 \end{matrix} \right\| = \left\| \begin{matrix} k13 & k33 & k53 \\ p13 & p33 & p53 \\ l13 & l33 & l53 \end{matrix} \right\| \quad (8)$$

$$\begin{Vmatrix} c1+c2 & c3+c4 \\ d1+d2 & d3+d4 \end{Vmatrix} * \begin{Vmatrix} a22-a21 & a24-a23 & a25 \\ a42-a41 & a44-a43 & a45 \end{Vmatrix} = \begin{Vmatrix} k23-k13 & k43-k33 & m53 \\ p23-p13 & p43-p33 & n53 \end{Vmatrix}$$
(9)

Aus den Beziehungen (8) und (9) erhält man die Matrix $L_4$

Die 100 Elementarmultiplikationen setzen sich wie folgt zusammen:

| Formel | | Multiplikationen | |
|---|---|---|---|
| (1) | 3x3 * 3x2 | 15 | Hopcroft |
| (2) | 3x3 * 3x2 | 14 | Kemper |
| (4) | 3x3 * 3x3 | 23 | Ladermann/Sýkora |
| (5) | 2x3 * 3x3 | 15 | Hopcroft |
| (6) | 2x2 * 2x2 | 7 | Strassen |
| (8) | 3x2 * 2x3 | 15 | Hopcroft |
| (9) | 2x2 * 2x3 | 11 | Lt. Bini evtl. nur 10?!? |
| Summe | | 100 | |

Sobald die verwendeten Einzelverfahren verbessert werden, sinkt die Anzahl der Elementarmultiplikationen unter 100. Es wäre daher zu überlegen, die Fälle „3x3 * 3x2“, „3x2 * 2x3“ und „2x2 * 2x3“ neu anzugehen. Vielleicht sind dort die Erfolgsaussichten größer als beim Ladermann/Sýkora-Fall „3x3 * 3x3“.

Laut Valery B. Alekseyev “On the complexity of some algorithms of matrix multiplication” [1](1982) benötigt die 2x2x3-Multiplikation mindestens 11 Elementarmultiplikationen. Das widerspricht dem Hinweis, es könne auch mit 10 Elementarmultiplikationen gehen.

Bisher habe ich in der Literatur kein Verfahren gefunden, das den Fall (2) 3x3 * 3x2 mit 14 Elementarmultiplikationen bewältigt. Für den hier vorliegenden Sonderfall $a_{33} = 0$ habe ich am 4.11.2012 per SAT Solver Z3 so ein Verfahren gefunden. Es ist in dieser Unterlage auf Seite 6 zu finden.

[1] *“It is shown that the multiplicative complexity of multiplication of a 3 × 2 matrix by a 2 × 2 matrix is equal to 11 for arbitrary field of constants.”*

## Yacas-Skript des Verfahrens

```
#  Script for Macarov's 5x5 Matrix Multiplication L = B * A

#  force creation of function at the beginning
#  as it is not possible at the end. Bug?
Simplify(1+1);

# Matrix B
B :=
{{b1,b2,b3,b4,b5},{c1,c2,c3,c4,c5},{g1,g2,g3,g4,g5},{d1,d2,d3,
d4,d5},{r1,r2,r3,r4,r5}};

# Matrix A
# (with uppercase "A" elements to avoid clashes)
A :=
{{A11,A12,A13,A14,A15},{A21,A22,A23,A24,A25},{A31,A32,A33,A34,
A35},{A41,A42,A43,A44,A45},{A51,A52,A53,A54,A55}};


products := 0;

#  Fomula (1)

a11 := b1 + b2 - c1 - c2;
a12 := b3 + b4 - c3 - c4;
a13 := b5;
a21 := g1 + g2 - d1 - d2;
a22 := g3 + g4 - d3 - d4;
a23 := g5;
a31 := r1 +r2;
a32 := r3 + r4;
a33 := r5;

#  "a" elements are written here as "A"
#  to distinguish from the "a" of the standard solution
b11 := A12;
b12 := A14;
b21 := A32;
b22 := A34;
b31 := A52;
b32 := A54;

#
#  3x3 * 3x2 Solution of Hopcroft 1973
```

```
#
#  note upper-case "P"!
P1 := (a11-a13-a21)*b11;
P2 := (a11-a12-a31)*(b11+b12);
P3 := (-a12+a22-a23)*b21;
P4 := (-a21+a22-a32)*b22;
P5 := (-a13-a32+a33)*(b31+b32);
P6 := (-a23-a31+a33)*b32;
P7 := a12*(b11+b12+b21+b22);
P8 := (-a12+a21)*(b11+b12+b22);
P9 := (-a21)*(b12+b22);
P10 := a32*(b21+b22+b31+b32);
P11 := (a23-a32)*(b21+b31+b32);
P12 := -a23*(b21+b31);
P13 := (-a13+a31)*(b11-b32);
P14 := (-a31)*(b12+b32);
P15 := a13*(b11+b31);

products := products + 15;

k21  := P1+P7+P8+P9+P15;
k41  := -P1+P2-P8-P9+P13-P14;
p21  := P3+P7+P8+P9-P12;
p41  := P4-P9+P10+P11+P12;
l211 := P5-P6-P11-P12+P13+P15;
l411 := P6+P10+P11+P12-P14;


#  Formula (2)

a11 := b2 - c2;
a12 := b4 - c4;
a13 := c5;
a21 := g2 - d2;
a22 := g4 - d4;
a23 := d5;
a31 := -r2;
a32 := -r4;
a33 := 0;

b11 := A12 - A22;
b12 := A14 - A24;
b21 := A32 - A42;
b22 := A34 - A44;
b31 := A52;
```

```
b32 := A54;

#
#  3x3 * 3x2 Solution for a33==0 of Kemper (04-Nov-2012)
#

P01 := (a22) * (b12-b22);
P02 := (a11-a23) * (-b12-b31);
P03 := (a31+a32) * (-b21);
P04 := (a12) * (-b21+b31);
P05 := (a11-a12-a13) * (-b31);
P06 := (a13-a23) * (-b31+b32);
P07 := (a12+a22+a32) * (b21-b22);
P08 := (-a21-a22+a23) * (b12);
P09 := (-a31) * (b11-b21);
P10 := (-a23) * (-b12-b32);
P11 := (a12) * (-b22+b31);
P12 := (-a11) * (-b11-b31);
P13 := (-a22+a31) * (b12-b21);
P14 := (a11-a21-a31) * (b11-b12);

products := products + 14;

m21 := -P04+P05+P12;
m41 := -P02+P05+P06+P10-P11;
n21 := P02-P08+P09+P12+P13-P14;
n41 := -P01-P08+P10;
l212 := -P03-P09;
l412 := P01-P03-P04-P07+P11+P13;


#  Formula (3)

l21 := l211 + l212;
l41 := l411 + l412;

#  Formula (4)

a11 := b1;
a12 := b3;
a13 := b5;
a21 := g1;
a22 := g3;
a23 := g5;
a31 := r1;
```

```
a32 := r3;
a33 := r5;

b11 := A11;
b12 := A13;
b13 := A15;
b21 := A31;
b22 := A33;
b23 := A35;
b31 := A51;
b32 := A53;
b33 := A55;

#
#  derived from 3x3 Solution of Ondrej Sýkora
#

P1 := a11 * b12;
P2 := a12 * b22;
P3 := a13 * b32;
P4 := (-a11+a21+a23) * (b12+b31-b11);
P5 := (-a12+a22+a23) * (b22+b33-b23);
P6 := (-a12+a32+a33) * (b22+b31-b21);
P7 := (-a11+a31+a33) * (b12+b33-b13);
P8 := (a21+a23) * (b11-b12);
P9 := (a22+a23) * (b23-b22);
P10 := (a32+a33) * (b21-b22);
P11 := (a31+a33) * (b13-b12);
P12 := (a11-a21) * (b31-b11);
P13 := (a12-a22) * (b33-b23);
P14 := (a12-a32) * (b31-b21);
P15 := (a11-a31) * (b33-b13);
P16 := a22 * b21;
P17 := a21 * b13;
P18 := a31 * b11;
P19 := a32 * b23;
P20 := (a11 + a12 + a13 -a23 - a21 - a33 - a32) * b31;
P21 := (a11 + a12 + a13 -a23 - a22 - a33 - a31) * b33;
P22 := a23 * (b32 - b12 - b22 + b11 - b31 + b23 - b33);
P23 := a33 * (b32 - b12 - b22 + b21 - b31 + b13 - b33);

k12 := P20 + P4 + P8 + P1 + P6 + P10 + P2;
k32 := P1 + P2 + P3;
k52 := P21 + P5 + P9 + P2 + P7 + P11 + P1;
p12 := P1 + P4 + P8 + P12 + P16;
```

```
p32 := P22 + P4 + P12 + P1 + P5 + P13 + P2;
p52 := P2 + P5 + P9 + P13 + P17;
l12 := P2 + P6 + P10 + P14 + P18;
l32 := P23 + P6 + P14 + P2 + P7 + P15 + P1;
l52 := P1 + P7 + P11 + P15 + P19;

products := products + 23;


#  Formula (5)(7a)(7b)

a11 := c1;
a12 := c3;
a13 := c5;
a21 := d1;
a22 := d3;
a23 := d5;

b11 := A11 - A21 - A12 + A22;   // v1
b12 := A13 - A23 - A14 + A24;   // v2
b13 := A15 - A25;
b21 := A31 - A41 - A32 + A42;   // v3
b22 := A33 - A43 - A34 + A44;   // v4
b23 := A35 - A45;
b31 := A51 - A52;
b32 := A53 - A54;
b33 := A55;

#
#  2x3 * 3x3 Solution of Hopcroft 1973
#  (transposed from 3x3 * 3x2)
#

P1 := a11*(b11-b31-b12);
P2 := (a11+a21)*(b11-b21-b13);
P3 := a12*(-b21+b22-b32);
P4 := a22*(-b12+b22-b23);
P5 := (a13+a23)*(-b31-b23+b33);
P6 := a23*(-b32-b13+b33);
P7 := (a11+a21+a12+a22)*b21;
P8 := (a11+a21+a22)*(-b21+b12);
P9 := (a21+a22)*(-b12);
P10 := (a12+a22+a13+a23)*b23;
P11 := (a12+a13+a23)*(b32-b23);
P12 := (a12+a13)*(-b32);
```

```
P13 := (a11-a23)*(-b31+b13);
P14 := (a21+a23)*(-b13);
P15 := (a11+a13)*b31;

f511 := P1+P7+P8+P9+P15;        //  m12 - k22
f512 := P3+P7+P8+P9-P12;        //  m32 - k42
m52  := P5-P6-P11-P12+P13+P15;
f521 := -P1+P2-P8-P9+P13-P14;   //  n12 - p22
f522 := P4-P9+P10+P11+P12;      //  n32 - p42
n52  := P6+P10+P11+P12-P14;

products := products + 15;

#  Matrix L3
m321 := f511;  // m12 - k22
m323 := f512;  // m32 - k42
m341 := f521;  // n12 - p22
m343 := f522;  // n32 - p42


#  Formula (6)

a11 := b2 - c1 - c2;
a12 := b4 - c3 - c4;
a21 := g2 - d1 - d2;
a22 := g4 - d3 - d4;

b11 := A22 - A12 - A21;
b12 := A24 - A23 - A14;
b21 := A42 - A41 - A32;
b22 := A44 - A43 - A34;


#  2x2 Strassen

P1 := (a12 - a22) * (b21 + b22);
P2 := (a11 + a22) * (b11 + b22);
P3 := (a11 - a21) * (b11 + b12);
P4 := (a11 + a12) * b22;
P5 := a11 * (b12 - b22);
P6 := a22 * (b21 - b11);
P7 := (a21 + a22) * b11;

products := products + 7;
```

```
k22 := P1 + P2 - P4 + P6;
k42 := P4 + P5;
p22 := P6 + P7;
p42 := P2 - P3 + P5 - P7;


#  Formula (7a+b)
#  --> Formula (1)

#  Formula (8)
a11 := b2;
a12 := b4;
a21 := g2;
a22 := g4;
a31 := r2;
a32 := r4;

b11 := A21;
b12 := A23;
b13 := A25;
b21 := A41;
b22 := A43;
b23 := A45;

#
#  3x2 * 2x3 Solution of Hopcroft 1973
#

P1 := (a11-a12)*b11;
P2 := a12*(b11+b21);
P3 := a21*b12;
P4 := a22*b22;
P5 := a31*(b13+b23);
P6 := (-a31+a32)*b23;
P7 := (a11+a21)*(b11+b12+b21+b22);
P8 := (a11-a12+a21)*(b11+b21+b22);
P9 := (a11-a12+a21-a22)*(b21+b22);
P10 := (a22+a32)*(b12+b13+b22+b23);
P11 := (a22-a31+a32)*(b12+b13+b23);
P12 := (-a21+a22-a31+a32)*(b12+b13);
P13 := (a12+a31)*(b11-b23);
P14 := (-a12-a32)*(b21+b23);
P15 := (a11+a31)*(b11+b13);

products := products + 15;
```

```
k13 := P1 + P2;
k33 := - P2 - P3 + P7 - P8;
k53 := - P1 - P5 - P13 + P15;
p13 := - P1 - P4 + P8 - P9;
p33 := P3 + P4;
p53 := - P3 - P6 + P11 - P12;
l13 := -P2 - P6 + P13 - P14;
l33 := - P4 - P5 + P10 - P11;
l53 := P5 + P6;


#  Formula (9)

a11 := c1 + c2;
a12 := c3 + c4;
a21 := d1 + d2;
a22 := d3 + d4;

b11 := A22 - A21;
b12 := A24 - A23;
b13 := A25;
b21 := A42 - A41;
b22 := A44 - A43;
b23 := A45;

#
#  2x2 * 2x3  Solution
#
P1 := (a12 - a22) * (b21 + b22);
P2 := (a11 + a22) * (b11 + b22);
P3 := (a11 - a21) * (b11 + b12);
P4 := (a11 + a12) * b22;
P5 := a11 * (b12 - b22);
P6 := a22 * (b21 - b11);
P7 := (a21 + a22) * b11;
P8 := a11 * b13;
P9 := a12 * b23;
P10 := a21 * b13;
P11 := a22 * b23;

products := products + 11;

f911 := P1 + P2 - P4 + P6;  //  k23 - k13
f912 := P4 + P5;            //  k43 - k33
```

```
m53  := P8 + P9;
f921 := P6 + P7;            //  p23 - p13
f922 := P2 - P3 + P5 - P7;  //  p43 - p33
n53  := P10 + P11;

#  Matrix L4
m421 := -f911;  //  k13 - k23
m423 := -f912;  //  k33 - k43
m441 := -f921;  //  p13 - p23
m443 := -f922;  //  p33 - p43

#  Matrix L1
k23 := -m421 + k13;
k43 := -m423 + k33;
p23 := -m441 + p13;
p43 := -m443 + p33;

# L1:=  L2 +   L3 +   L4
k1   :=   0 +  k12 +  k13;
k2   := k21 +  k22 +  k23;
k3   :=   0 +  k32 +  k33;
k4   := k41 +  k42 +  k43;
k5   :=   0 +  k52 +  k53;
m121 :=   0 + m321 + m421;  //  m1 - m2
m2   := m21 +  k22 +  k23;
m123 :=   0 + m323 + m423;  //  m3 - m4
m4   := m41 +  k42 +  k43;
m5   :=   0 +  m52 +  m53;
p1   :=   0 +  p12 +  p13;
p2   := p21 +  p22 +  p23;
p3   :=   0 +  p32 +  p33;
p4   := p41 +  p42 +  p43;
p5   :=   0 +  p52 +  p53;
m141 :=   0 + m341 + m441;  // n1 - n2
n2   := n21 +  p22 +  p23;
m143 :=   0 + m343 + m443;  // n3 - n4
n4   := n41 +  p42 +  p43;
n5   :=   0 +  n52 +  n53;
l1   :=   0 +  l12 +  l13;
l2   := l21 +    0 +    0;
l3   :=   0 +  l32 +  l33;
l4   := l41 +    0 +    0;
l5   :=   0 +  l52 +  l53;

# Matrix L
```

```
m1 := m121 + m2;
m3 := m123 + m4;
n1 := m141 + n2;
n3 := m143 + n4;

#  using the original notation of Makarov
L :=
{{k1,k2,k3,k4,k5},{m1,m2,m3,m4,m5},{p1,p2,p3,p4,p5},{n1,n2,n3,
n4,n5},{l1,l2,l3,l4,l5}};

#  Check result
#  this takes a while ...
Simplify(L - B*A);

#  tell total number of products
products;
```

## Formulierung des Verfahrens in der üblichen Matrixnotation

Makarov hat in seinem Papier eine eigene Schreibweise für Matrizenelemente gewählt und statt A*B das Matrizenprodukt B*A berechnet. Wenn man das auf A*B umstellt und die Schreibweise mit Zeilen- und Spaltenindex verwendet, ergibt sich folgendes Verfahren:

```
#
#  Yacas script s5x5x5_100.Makarov.txt created 11 Nov 2012 17:56:48
#
#  Makarov solution
#
#  Fast matrix multiplication method for 5x5x5 matrices.
#
#  intermediate products: 100
#
#  literal 0 from mod2: 6613
#  literal 1 from mod2: 887
#  literal +1 from rev: 568
#  literal -1 from rev: 319

P01 := (a11+a12-a15-a21-a22-a31-a32+a41+a42) * (b12);
P02 := (a11+a12-a13-a14-a21-a22+a23+a24-a51-a52) * (b12+b14);
P03 := (-a13-a14+a23+a24+a33+a34-a35-a43-a44) * (b32);
P04 := (-a31-a32+a33+a34+a41+a42-a43-a44-a53-a54) * (b34);
P05 := (-a15-a53-a54+a55) * (b52+b54);
P06 := (-a35-a51-a52+a55) * (b54);
P07 := (a13+a14-a23-a24) * (b12+b14+b32+b34);
P08 := (-a13-a14+a23+a24+a31+a32-a41-a42) * (b12+b14+b34);
P09 := (-a31-a32+a41+a42) * (b14+b34);
P10 := (a53+a54) * (b32+b34+b52+b54);
P11 := (a35-a53-a54) * (b32+b52+b54);
P12 := (-a35) * (b32+b52);
```

```
P13 := (-a15+a51+a52) * (b12-b54);
P14 := (-a51-a52) * (b14+b54);
P15 := (a15) * (b12+b52);
P16 := (a34-a44) * (b14-b24-b34+b44);
P17 := (a12-a22-a45) * (-b14+b24-b52);
P18 := (a52+a54) * (b32-b42);
P19 := (a14-a24) * (-b32+b42+b52);
P20 := (-a12+a14+a22-a24+a25) * (b52);
P21 := (a25-a45) * (-b52+b54);
P22 := (a14-a24+a34-a44-a54) * (b32-b34-b42+b44);
P23 := (-a32-a34+a42+a44+a45) * (b14-b24);
P24 := (a52) * (b12-b22-b32+b42);
P25 := (a45) * (b14-b24+b54);
P26 := (a14-a24) * (-b34+b44+b52);
P27 := (-a12+a22) * (-b12+b22-b52);
P28 := (-a34+a44-a52) * (b14-b24-b32+b42);
P29 := (a12-a22-a32+a42+a52) * (b12-b14-b22+b24);
P30 := (a11) * (b13);
P31 := (a13) * (b33);
P32 := (a15) * (b53);
P33 := (-a11+a31+a35) * (-b11+b13+b51);
P34 := (-a13+a33+a35) * (b33-b35+b55);
P35 := (-a13+a53+a55) * (-b31+b33+b51);
P36 := (-a11+a51+a55) * (b13-b15+b55);
P37 := (a31+a35) * (b11-b13);
P38 := (a33+a35) * (-b33+b35);
P39 := (a53+a55) * (b31-b33);
P40 := (a51+a55) * (-b13+b15);
P41 := (a11-a31) * (-b11+b51);
P42 := (a13-a33) * (-b35+b55);
P43 := (a13-a53) * (-b31+b51);
P44 := (a11-a51) * (-b15+b55);
P45 := (a33) * (b31);
P46 := (a31) * (b15);
P47 := (a51) * (b11);
P48 := (a53) * (b35);
P49 := (a11+a13+a15-a31-a35-a53-a55) * (b51);
P50 := (a11+a13+a15-a33-a35-a51-a55) * (b55);
P51 := (a35) * (b11-b13-b33+b35-b51+b53-b55);
P52 := (a55) * (-b13+b15+b31-b33-b51+b53-b55);
P53 := (a21) * (b11-b12-b13+b14-b21+b22+b23-b24-b51+b52);
P54 := (a21+a41) * (b11-b12-b15-b21+b22+b25-b31+b32+b41-b42);
P55 := (a23) * (-b31+b32+b33-b34+b41-b42-b43+b44-b53+b54);
P56 := (a43) * (-b13+b14+b23-b24+b33-b34-b35-b43+b44+b45);
P57 := (a25+a45) * (-b35+b45-b51+b52+b55);
P58 := (a45) * (-b15+b25-b53+b54+b55);
P59 := (a21+a23+a41+a43) * (b31-b32-b41+b42);
P60 := (a21+a41+a43) * (b13-b14-b23+b24-b31+b32+b41-b42);
P61 := (a41+a43) * (-b13+b14+b23-b24);
P62 := (a23+a25+a43+a45) * (b35-b45);
P63 := (a23+a25+a45) * (-b35+b45+b53-b54);
P64 := (a23+a25) * (-b53+b54);
P65 := (a21-a45) * (b15-b25-b51+b52);
P66 := (a41+a45) * (-b15+b25);
P67 := (a21+a25) * (b51-b52);
P68 := (a14-a23-a24-a34+a43+a44) * (-b32-b34-b41+b42-b43+b44);
P69 := (a12-a21-a22+a34-a43-a44) * (-b12-b21+b22-b34-b43+b44);
P70 := (a12-a21-a22-a32+a41+a42) * (-b12-b14-b21+b22-b23+b24);
P71 := (a12+a14-a21-a22-a23-a24) * (-b34-b43+b44);
P72 := (a12-a21-a22) * (-b14-b23+b24+b34+b43-b44);
P73 := (a34-a43-a44) * (b12+b21-b22-b32-b41+b42);
P74 := (a32+a34-a41-a42-a43-a44) * (-b12-b21+b22);
```

```
P75 := (a12-a14) * (b21);
P76 := (a14) * (b21+b41);
P77 := (a32) * (b23);
P78 := (a34) * (b43);
P79 := (a52) * (b25+b45);
P80 := (-a52+a54) * (b45);
P81 := (a12+a32) * (b21+b23+b41+b43);
P82 := (a12-a14+a32) * (b21+b41+b43);
P83 := (a12-a14+a32-a34) * (b41+b43);
P84 := (a34+a54) * (b23+b25+b43+b45);
P85 := (a34-a52+a54) * (b23+b25+b45);
P86 := (-a32+a34-a52+a54) * (b23+b25);
P87 := (a14+a52) * (b21-b45);
P88 := (-a14-a54) * (b41+b45);
P89 := (a12+a52) * (b21+b25);
P90 := (a23+a24-a43-a44) * (-b41+b42-b43+b44);
P91 := (a21+a22+a43+a44) * (-b21+b22-b43+b44);
P92 := (a21+a22-a41-a42) * (-b21+b22-b23+b24);
P93 := (a21+a22+a23+a24) * (-b43+b44);
P94 := (a21+a22) * (-b23+b24+b43-b44);
P95 := (a43+a44) * (b21-b22-b41+b42);
P96 := (a41+a42+a43+a44) * (-b21+b22);
P97 := (a21+a22) * (b25);
P98 := (a23+a24) * (b45);
P99 := (a41+a42) * (b25);
P100 := (a43+a44) * (b45);

c11 := P30+P31+P33+P35+P37+P39+P49+P75+P76;
c12 := P01+P07+P08+P09+P15+P68+P69-P71+P73+P75+P76+P90+P91-P93+P95;
c13 := P30+P31+P32-P76-P77+P81-P82;
c14 := -P01+P02-P08-P09+P13-P14+P71+P72-P76-P77+P81-P82+P93+P94;
c15 := P30+P31+P34+P36+P38+P40+P50-P75-P79-P87+P89;
c21 := -P19+P20+P27+P53+P59+P60+P61+P67+P68+P69-P71+P73+P75+P76;
c22 := -P19+P20+P27+P68+P69-P71+P73+P75+P76+P90+P91-P93+P95;
c23 := -P17+P20+P21+P25-P26+P55+P59+P60+P61-P64+P71+P72-P76-P77+P81-P82;
c24 := -P17+P20+P21+P25-P26+P71+P72-P76-P77+P81-P82+P93+P94;
c25 := P57-P58-P63-P64+P65+P67+P97+P98;
c31 := P30+P33+P37+P41+P45-P75-P78+P82-P83;
c32 := P03+P07+P08+P09-P12+P73+P74-P75-P78+P82-P83+P95+P96;
c33 := P30+P31+P33+P34+P41+P42+P51+P77+P78;
c34 := P04-P09+P10+P11+P12+P69-P70+P72-P74+P77+P78+P91-P92+P94-P96;
c35 := P31+P34+P38+P42+P46-P77-P80+P85-P86;
c41 := P17-P23+P24+P27+P28-P29-P53+P54-P60-P61+P65-P66+P73+P74-P75-P78+P82-
P83;
c42 := P17-P23+P24+P27+P28-P29+P73+P74-P75-P78+P82-P83+P95+P96;
c43 := -P16-P23+P25+P56-P61+P62+P63+P64+P69-P70+P72-P74+P77+P78;
c44 := -P16-P23+P25+P69-P70+P72-P74+P77+P78+P91-P92+P94-P96;
c45 := P58+P62+P63+P64-P66+P99+P100;
c51 := P31+P35+P39+P43+P47-P76-P80+P87-P88;
c52 := P05-P06-P11-P12+P13+P15-P18-P24;
c53 := P30+P31+P35+P36+P43+P44+P52-P78-P79+P84-P85;
c54 := P06+P10+P11+P12-P14+P16-P18-P19-P22+P26+P28;
c55 := P30+P36+P40+P44+P48+P79+P80;

Simplify(c11 - (a11*b11+a12*b21+a13*b31+a14*b41+a15*b51));
Simplify(c12 - (a11*b12+a12*b22+a13*b32+a14*b42+a15*b52));
Simplify(c13 - (a11*b13+a12*b23+a13*b33+a14*b43+a15*b53));
Simplify(c14 - (a11*b14+a12*b24+a13*b34+a14*b44+a15*b54));
Simplify(c15 - (a11*b15+a12*b25+a13*b35+a14*b45+a15*b55));
Simplify(c21 - (a21*b11+a22*b21+a23*b31+a24*b41+a25*b51));
Simplify(c22 - (a21*b12+a22*b22+a23*b32+a24*b42+a25*b52));
Simplify(c23 - (a21*b13+a22*b23+a23*b33+a24*b43+a25*b53));
```

```
Simplify(c24 - (a21*b14+a22*b24+a23*b34+a24*b44+a25*b54));
Simplify(c25 - (a21*b15+a22*b25+a23*b35+a24*b45+a25*b55));
Simplify(c31 - (a31*b11+a32*b21+a33*b31+a34*b41+a35*b51));
Simplify(c32 - (a31*b12+a32*b22+a33*b32+a34*b42+a35*b52));
Simplify(c33 - (a31*b13+a32*b23+a33*b33+a34*b43+a35*b53));
Simplify(c34 - (a31*b14+a32*b24+a33*b34+a34*b44+a35*b54));
Simplify(c35 - (a31*b15+a32*b25+a33*b35+a34*b45+a35*b55));
Simplify(c41 - (a41*b11+a42*b21+a43*b31+a44*b41+a45*b51));
Simplify(c42 - (a41*b12+a42*b22+a43*b32+a44*b42+a45*b52));
Simplify(c43 - (a41*b13+a42*b23+a43*b33+a44*b43+a45*b53));
Simplify(c44 - (a41*b14+a42*b24+a43*b34+a44*b44+a45*b54));
Simplify(c45 - (a41*b15+a42*b25+a43*b35+a44*b45+a45*b55));
Simplify(c51 - (a51*b11+a52*b21+a53*b31+a54*b41+a55*b51));
Simplify(c52 - (a51*b12+a52*b22+a53*b32+a54*b42+a55*b52));
Simplify(c53 - (a51*b13+a52*b23+a53*b33+a54*b43+a55*b53));
Simplify(c54 - (a51*b14+a52*b24+a53*b34+a54*b44+a55*b54));
Simplify(c55 - (a51*b15+a52*b25+a53*b35+a54*b45+a55*b55));

#  No linear dependencies between intermediate products found

#  add operations: 404
#  sub operations: 258

#
#  end of 5x5x5 solution s5x5x5_100.Makarov.txt
#
```

## Aus dem Originalpapier extrahiert und übersetzt

Матрицы S1 и S2 из (13) имеют вид (11). Следовательно. при умножении по формуле (10) матриц S1 и S2 на вектор достаточно выполнить 5 + 5=10 умножений.
При умножении матрицы S3 на вектор достаточно выполнить 4 умножения. Таким образом, задачу 5 можно выполнить, используя 14 умножений.

Matrices S1 and S2 (13) have the form (11). Consequently. when multiplied by (10) the matrices S1 and S2 is sufficient to perform the vector 5 + 5 = 10 multiplications.
When multiplied by the vector matrix S3 is sufficient to perform four multiplications. So way, task 5 can be made using 14 multiplications.

Алгоритм вычисления (2), требующий въшолнения 14 умножений. таков.
Рассмотрим задачу умножения вектора на матрицу Q вида

Разложим матрицу Q в сумму 6 матриц единичного ранга:

где a = q5 - q2 -q4, Ь = 1 + a. Согласно (10). для умножения вектора на матрицу Q достаточно въшолнить 6 умножений.

Ив (10) также следует. что для умножения вектора на матрицу Q1, состоящую нз первых пяти слагаемых суммы (10), т. е.

достаточно выполнить 5 умножений.
Представим (2) в виде умножений матрицы

Представим (12) в виде

Algorithm to compute (2), which requires vsholneniya 14 multiplications. is as follows.
Consider the problem of multiplying a vector by a matrix of the form Q

We decompose the matrix Q in the amount of 6 matrix of unit rank:

where a = q5 - q2-q4, b = 1 + a. According to (10). to multiply a vector by a matrix Q enough vsholnit 6 multiplications.

In (10) as follows. that the multiplication of a vector by a matrix Q1, NC consisting of the first five terms in the sum (10), ie,

just follow five multiplications.
Represent (2) as a matrix multiplication  Represent (12) as