```
restart
with(ArrayTools), with(ListTools), with(LinearAlgebra) :
dim := 4 :
id := IdentityMatrix(dim) + Matrix(dim) : idd := IdentityMatrix(dim^2) + Matrix(dim^2) :
iddd := IdentityMatrix(dim^3) + Matrix(dim^3) :
ID := IdentityMatrix(2 dim) + Matrix(2 dim) :
IDD := IdentityMatrix((2 dim)^2) + Matrix((2 dim)^2) : IDDD := IdentityMatrix((2 dim)^3)
    + Matrix((2 dim)^3) :
P := Matrix(64, storage = sparse) :
for i to 8 do
    for j to 8 do
        P(8 (j - 1) + i, 8 (i - 1) + j) := 1 :
    end do
end do
tr2 := proc(x)
    local a, b, A, i, j, k;
    a := Size(x, 1) / dim;
    b := Size(x, 2) / dim;
    A := Matrix(a, b, storage = 'sparse');
    for i to a do
        for j to b do
            for k to dim do
                A(i, j) := A(i, j) + x(dim * (i - 1) + k, dim * (j - 1) + k);
            end do;
        end do;
    end do;
    A := smap(simplify, A / dim);
    return A;
end proc:
spmm := proc(A::Matrix, B::Matrix)
    local m, n, Ae, Be, Bi, R, l, e, i;
    n, m := op(1, A);
    i, l := op(1, B);
    if i <> m then
        error "incompatible dimensions"
    end if;
    Ae := op(2, A);
    Be := op(2, B);
    R := Matrix(n, l, storage = sparse);
    for i to l do
        Bi, Be := selectremove(type, Be, (anything, i) = anything);
        Bi := map2(op, [1, 1], Bi);
        for e in Ae do
            n, m := op(1, e);
            if member(m, Bi) then
                R[n, i] := A[n, m] * B[m, i] + R[n, i] :
            end if;
        end do;
    end do;
    return R;
```

```
end proc:
smap := proc(f, A::{Matrix, Vector})
    local B, Ae, e;
    if A::Vector then
        B := Vector(op(1, A), storage = sparse);
    else
        B := Matrix(op(1, A), storage = sparse);
    end if;
    Ae := op(2, A);
    for e in Ae do
        B[op(1, e)] := f(op(2, e), args[3 ..nargs]);
end do;
return B;
 end proc:
ssimplify := x→smap(simplify, x) :
KP := proc(A::Matrix, B::Matrix)
    local m, n, Ae, Be, Ai, R, l, e, i, j, s, t;
    m, n := op(1, A);
    s, t := op(1, B);
    Ae := op(2, A);
    Be := op(2, B);
    R := Matrix(m * s, n * t, storage = sparse);
    for j to n do
        Ai, Ae := selectremove(type, Ae, (anything, j) = anything);
        Ai := map2(op, [1, 1], Ai);
        for e in Be do
            m, n := op(1, e);
            for i in Ai do
                R[(i − 1) * s + m, (j − 1) * t + n] := A[i, j] * B[m, n];
            end do;
        end do;
    end do;
    return R;
end proc:
check := proc(x)
    local A;
    A := convert(smap(simplify, x), Matrix);
    if Equal(A, Matrix(Size(x, 1), Size(x, 2))) then
        Good
    ;else
        Bad, A;
    end if
;end proc:
br := (x, y) → `.`(x, y) − `.`(y, x) :
```

$$fl := x \to -\frac{I\left(x - \frac{1}{x}\right)}{2} :$$

$$q := x \to \mathrm{e}^{\frac{1\,I\mathrm{Pi}\,x}{2}} :$$

$$F1 := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} : E1 := \begin{bmatrix} 0 & fl(t1) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & fl(I \cdot t1) & fl(t1) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & fl(t1) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -fl(I \cdot t1) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & fl(t1) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} :$$

$$K1 := \begin{bmatrix} t1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -t1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & t1\,I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -t1\,I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -t1\,I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & t1\,I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & t1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -t1 \end{bmatrix} :$$

$$F2 := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} : E2 := \begin{bmatrix} 0 & 0 & fl(t2) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & fl(t2) & fl(I \cdot t2) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -fl(I \cdot t2) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & fl(t2) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & fl(t2) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} :$$

$$K2 := \begin{bmatrix} t2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & t2\,I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -t2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -t2\,I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -t2\,I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & t2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & t2\,I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -t2 \end{bmatrix} :$$

$E3 := -(E1 \cdot E2 + I \cdot E2 \cdot E1) : F3 := -(F2 \cdot F1 - I \cdot F1 \cdot F2) :$

$H1 :=$

$$\begin{bmatrix} \text{lambda} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \text{lambda}-2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \text{lambda}+1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \text{lambda}-1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \text{lambda}-1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \text{lambda}-3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \text{lambda} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \text{lambda}-2 \end{bmatrix}:$$

$$H2 := \begin{bmatrix} \text{mu} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \text{mu}+1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \text{mu}-2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \text{mu}-1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \text{mu}-1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \text{mu} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \text{mu}-3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \text{mu}-2 \end{bmatrix}:$$

$check(br(E1, F1) - fl(K1)), check(br(E2, F2) - fl(K2)), check(br(E1, F2)), check(br(E2, F1)),$
$check(K1 \cdot F1 + F1 \cdot K1), check(K2 \cdot F2 + F2 \cdot K2), check(K1 \cdot E1 + E1 \cdot K1), check(K2 \cdot E2 + E2 \cdot K2),$
$check(K1 \cdot F2 - I \cdot F2 \cdot K1), check(K2 \cdot F1 - I \cdot F1 \cdot K2), check(K1 \cdot E2 + I \cdot E2 \cdot K1), check(K2 \cdot E1 + I \cdot E1 \cdot K2), check(E1^2), check(F1^2), check(E2^2), check(F2^2), check(E3^2), check(F3^2),$
$check(br(H1, E1) - 2 \cdot E1), check(br(H2, E1) + E1), check(br(H1, E2) + E2), check(br(H2, E2) - 2 \cdot E2), check(br(H1, F1) + 2 \cdot F1), check(br(H2, F1) - F1), check(br(H1, F2) - F2),$
$check(br(H2, F2) + 2 \cdot F2);$

Good, Good **(1)**