matrix and otherwise, it recurses through the above expression. The function is now *almost* complete:

```
   elim:{[h M];h::*M::x@>*'x;
          :[2>#x;x
           ;(,h),0,:\.f({1_x}'{x-h*(*x)%*h}'1_M)]}
   elim(M)
[[1 3   -4   8  ]
 [0 2.0 -2.0 6.0]
 [0 0    2.0 4.0]]
```

There are two corner cases to consider, though: (1) the matrix passed to $elim$ may contain negative coefficients, so the function may order the rows in such a way that a division by zero can happen, and (2) there can be zero coefficients in random places in the matrix, which could also cause a division by zero.

Problem (1) is easily remedied by ordering the rows by their absolute values, i.e. by defining M as

```
x@>*'#''x
```

instead of

```
x@>*'x
```

where `#''x` [Size-Each-Each] computes the absolute value of the matrix x.

The solution to (2) is a bit more complicated as it involves finding the first non-zero coefficient in a given row. Finding the first non-zero element of the vector h can be accomplished using the expression

```
d::*(~h)?0
```

which locates the first zero in the complement of h (and also binds it to the variable d, the divisor index).

Now we also have to cover the case where there is no suitable divisor, i.e. all elements of the row are zero. This does not necessarily indicate a singularity, because this case may occur in a recursive application of $elim$.

So we skip the division in the case where no non-zero column

exists by replacing the expression

```
{x-h*(*x)%*h}
```

by

```
:[d~[];x;x-h*(x@d)%h@d]
```

These two modifications result in the final *elim* function:

```
elim::{[h M d];h::*M::x@>*'#''x;d::*(~h)?0;
       :[2>#x;x;(,h),0,:\.f({1_x}'{:[d~[];x
       ;x-h*(x@d)%h@d]}'1_M)]}
```

The next step is called *substitution*. It will fill in the known variables in the rows with unknown variables in order to produce more known variables. At the end of the substitution step, the unaugmented part of the matrix (without the rightmost column) will be in diagonal form and the rightmost column will contain the values of the variables.

It is easier to start from the top/left, so the matrix is flipped both horizontally and vertically first. The known variable is then in the leftmost column of the top row:

$$\begin{bmatrix} ax_3 & a & 0 & 0 \\ t & bx_3 & u & 0 \\ v & cx_3 & dx_2 & w \end{bmatrix}$$

The value of x_3 can now be computed by dividing ax_3 by a. As soon as x_3 is known, it can be inserted into the row below. At that point,

$$x_2 = \frac{t - bx_3}{u}$$

can be computed and, in turn,

$$x_1 = \frac{v - cx_3 - dx_2}{w}$$

can be calculated, thereby solving the entire system of equations.

Flipping a matrix both horizontally and vertically is done by the idiom Reverse-Reverse-Each: