

TMA4280: Introduction to Supercomputing

Suggested solutions

Problem set 1

January 2012

©Einar M. Rønquist

Department of Mathematical Sciences

NTNU, N-7491 Trondheim, Norway

All rights reserved

Exercise 1

7 digits (single precision).

Exercise 2

$$\begin{aligned}
 4.25 &= 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} \\
 \Rightarrow (4.25)_{10} &= (100.01)_2 \\
 &= (1.0001)_2 \cdot 2^2
 \end{aligned}$$

Comparing this with (3)-(5) in the notes, we identify

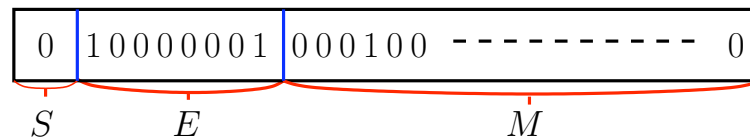
$$\begin{aligned}
 S &= 0 \\
 E - B = 2 &\Rightarrow E = B + 2 = 127 + 2 = 129
 \end{aligned}$$

But

$$\begin{aligned}
 E &= (129)_{10} = (128 + 1)_{10} = (10000001)_2 \\
 M &= (1.0001)_2
 \end{aligned}$$

where the leading bit of M is implicit in the representation.

Hence, the floating point representation of 4.25 is

**Exercise 3**

Relative accuracy of a floating point number in double precision is $2^{-52} = 2.2 \cdot 10^{-16}$. Hence, we have 16 digits of accuracy.

Exercise 4

One alternative is to use a double loop where the inner loop only goes up to approximately 10^9 .

Note that the data type `int` in C may only be in the $\pm 2^{15}$ (16 bits of information), although many newer platforms will use 32 bits to store integer information. The data type `long` will certainly use 32 bits for integer representation. Note that C also has the data type `long long` which uses 64 bits for integer representation. This alternative should certainly be enough as the range is $\pm 2^{63} \approx 9 \cdot 10^{18}$.

Exercise 5

For the first case we need n additions and n multiplications:

$$\begin{aligned}\underline{z} &= \underline{x} + c\underline{y} \\ \Downarrow \\ \mathcal{N}_{\text{ops}} &= n + n = 2n\end{aligned}$$

For the matrix-vector product, the computation of each component in \underline{y} requires n multiplications and $n - 1$ additions:

$$\begin{aligned}\underline{y} &= \underline{A}\underline{x} \\ \Downarrow \\ \mathcal{N}_{\text{ops}} &= n(n + (n - 1)) = n(2n - 1) \underset{\substack{\uparrow \\ n \gg 1}}{\simeq} 2n^2 = \mathcal{O}(n^2).\end{aligned}$$

Exercise 6

Solve $\underline{A}\underline{x} = \underline{b}$.

Total storage requirement

$$\left(\underbrace{n^2}_{\underline{A}} + \underbrace{n}_{\underline{x}} + \underbrace{n}_{\underline{b}} \right) \cdot 8 \text{ bytes.}$$

Assuming that $n \gg 1$, this is approximately equal to $8n^2$ bytes.

Constraint:

$$\begin{aligned}8n^2 &< 1 \cdot 10^9 \\ \Rightarrow \quad n &\leq 11000.\end{aligned}$$

Hence, we can only solve a system with approximately 10^4 unknowns.

Code

```
#include <stdio.h>

const double A[][3] = {{0.3, 0.4, 0.3},
                        {0.7, 0.1, 0.2},
                        {0.5, 0.5, 0.0}};

const double x[3] = {1.0, 1.0, 1.0};

int main(int argc, char** argv)
{
    double y[3];
    int i, j;
    for (i=0; i < 3; ++i) {
        y[i] = 0.0;
        for (j=0; j < 3; ++j)
            y[i] += A[i][j]*x[j];
    }
    printf("result: y=%f %f %f\n", y[0], y[1], y[2]);
    return 0;
}
```