CS/SE 4X03 — Assignment 1

21 September, 2021

Due date: 1 October

Instructions

- If you write your solutions by hand, please ensure your handwriting is legible. We may subtract marks for hard-to-read solutions.
- Submit to Avenue a **PDF file** containing your solutions and the **required** MATLAB **files**.

  Assignments in other formats, e.g. IMG, PNG, **will not be marked**.

  Name your MATLAB files **exactly** as specified.
- Name your PDF file **Lastname-Firstname-studentnumber.pdf**.
- Submit **only what is required**.
- Do not submit zipped files. We will **ignore any compressed file** containing your files.

Problem 1 [2 points]

Solution.

- a+b+c, (a+b)+c

  a+b+c evaluates to (a+b)+c, as the evaluation is from left to right. a+b evaluates to a. This can be checked by a+b==a, which gives true.

  (a+b)+c is the same as a+c.

  In decimal, a+c = -1.10714946411818e+17+1.107149464118181e+17 = 100, but MATLAB gives 96.

  If we output a and c to say 18 digits, e.g.

```
>> fprintf("%.18e\n",a)
-1.107149464118180000e+17
>> fprintf("%.18e\n",c)
1.107149464118180960e+17
```

  it appears a is the same after the conversion decimal to binary to decimal, while c is not. So a has roundoff in the conversion to binary. Then
  -1.107149464118180000e+17+1.107149464118180960e+17 gives 96

- a+(b+c)

  b+c is the same as c. Then this expression evaluates to a+c, which is 96 as above.

- (a+c)+b

  (a+c)+b evaluates to 96+b = 95.196036565001179

- a+(c+b) is the same as a+c.

Which is the most accurate result from abve (not required for this problem)? The exact a+c is 100. From it we subtract 0.839634349988262, which gives 99.160365650011738, so 96 is the more accurate one.

Problem 2 [7 points]

Solution.

a. Take e.g. $a = 5.001$ and $b = 5.002$.

$a + b = 5.001 + 5.002 = 10.003$ which after rounding is 1.000×10^1. Then $1.000 \times 10^1/2$ is 5.000.

b. $a = 1.234 \times 10^{10}$, $b = 5.555$, $c = 1.002 \times 10^{-5}$.

Then

$a * b = 6.8487 \times 10^{10}$ rounds to 6.849×10^{10}, and $(a*b)*c = 6.849 \times 10^{10} \times 1.002 \times 10^{-5} = 6.862\,698 \times 10^{5}$ rounds to 6.863×10^{5}.

$b*c = 5.566\,11 \times 10^{-5}$ rounds to 5.566×10^{-5} and $a*(b*c) = 1.234 \times 10^{10} \times 5.566 \times 10^{-5} = 6.868\,444 \times 10^{5}$ rounds to 6.868×10^{5}

c. $a = 1$, $b = 2.000 \times 10^{-4}$, $c = 4.000 \times 10^{-4}$

Then

$a + b = 1.0002$, which rounds to 1, and $(a + b) + c = 1 + 4.000 \times 10^{-5}$ rounds to 1.

$b + c = 6.000 \times 10^{-4}$ and $a + (b + c) = 1.0000 + 6.000 \times 10^{-4} = 1.0006$ rounds to 1.001

d. If no overflow occurs

$$\begin{aligned} 2a \leq a + b &\leq 2b \\ 2a = \mathrm{fl}\,(2a) \leq \mathrm{fl}\,(a + b) &\leq \mathrm{fl}\,(2b) = 2b \\ a \leq \mathrm{fl}\,(a + b)\,/2 = \mathrm{fl}\,((a + b)/2) &\leq b \end{aligned}$$

Problem 3 [8 points]

Solution.

expsum1

x	accurate	approx.	abs. error	rel. error
-20.0	2.061153622439e-09	5.621884472130e-09	3.56e-09	1.73e+00
-15.0	3.059023205018e-07	3.059094197302e-07	7.10e-12	2.32e-05
-10.0	4.539992976248e-05	4.539992962303e-05	1.39e-13	3.07e-09
-5.0	6.737946999085e-03	6.737946999084e-03	1.43e-15	2.13e-13
-1.0	3.678794411714e-01	3.678794411714e-01	1.11e-16	3.02e-16
1.0	2.718281828459e+00	2.718281828459e+00	0.00e+00	0.00e+00
5.0	1.484131591026e+02	1.484131591026e+02	5.68e-14	3.83e-16
10.0	2.202646579481e+04	2.202646579481e+04	7.28e-12	3.30e-16
15.0	3.269017372472e+06	3.269017372472e+06	9.31e-10	2.85e-16
20.0	4.851651954098e+08	4.851651954098e+08	5.96e-08	1.23e-16

expsum2

x	accurate	approx.	abs. error	rel. error
-20.0	2.061153622439e-09	2.061153622439e-09	4.14e-25	2.01e-16
-15.0	3.059023205018e-07	3.059023205018e-07	1.06e-22	3.46e-16
-10.0	4.539992976248e-05	4.539992976248e-05	1.36e-20	2.99e-16
-5.0	6.737946999085e-03	6.737946999085e-03	2.60e-18	3.86e-16

```
  -1.0   3.678794411714e-01   3.678794411714e-01     5.55e-17     1.51e-16
   1.0   2.718281828459e+00   2.718281828459e+00     0.00e+00     0.00e+00
   5.0   1.484131591026e+02   1.484131591026e+02     5.68e-14     3.83e-16
  10.0   2.202646579481e+04   2.202646579481e+04     7.28e-12     3.30e-16
  15.0   3.269017372472e+06   3.269017372472e+06     9.31e-10     2.85e-16
  20.0   4.851651954098e+08   4.851651954098e+08     5.96e-08     1.23e-16

expsum3
   x        accurate              approx.          abs. error    rel. error
 -20.0   2.061153622439e-09   8.940696716309e-08     8.73e-08     4.24e+01
 -15.0   3.059023205018e-07   3.057066351175e-07     1.96e-10     6.40e-04
 -10.0   4.539992976248e-05   4.539992369246e-05     6.07e-12     1.34e-07
  -5.0   6.737946999085e-03   6.737946999081e-03     4.32e-15     6.41e-13
  -1.0   3.678794411714e-01   3.678794411714e-01     0.00e+00     0.00e+00
   1.0   2.718281828459e+00   2.718281828459e+00     0.00e+00     0.00e+00
   5.0   1.484131591026e+02   1.484131591026e+02     5.68e-14     3.83e-16
  10.0   2.202646579481e+04   2.202646579481e+04     7.28e-12     3.30e-16
  15.0   3.269017372472e+06   3.269017372472e+06     9.31e-10     2.85e-16
  20.0   4.851651954098e+08   4.851651954098e+08     5.96e-08     1.23e-16
```

I will explain this output in class.

Problem 4 [6 points] 2.2 GHz Quad-Core Intel Core i7

```
Memory required:  3914K.


LINPACK benchmark
Single precision
Digits:  6
Array size 1000 X 1000.
Average rolled and unrolled performance:

    Reps Time(s) DGEFA   DGESL  OVERHEAD    MFLOPS      GFLOPS
----------------------------------------------------------------
      16   0.55  93.53%   0.77%   5.70%   5200.599       5.201
      32   1.11  93.50%   0.85%   5.65%   5119.632       5.120
      64   2.35  93.64%   0.83%   5.53%   4837.643       4.838
     128   4.76  93.56%   0.82%   5.62%   4779.691       4.780
     256   9.15  93.56%   0.79%   5.65%   4971.453       4.971
     512  19.06  93.63%   0.76%   5.61%   4771.622       4.772
```

```
Memory required:  7824K.


LINPACK benchmark
Double precision
Digits:  15
Array size 1000 X 1000.
Average rolled and unrolled performance:

    Reps Time(s) DGEFA   DGESL  OVERHEAD    MFLOPS      GFLOPS
----------------------------------------------------------------
      16   0.98  95.69%   0.84%   3.47%   2827.190       2.827
      32   1.97  95.66%   0.86%   3.48%   2824.665       2.825
      64   4.24  95.68%   0.88%   3.44%   2622.924       2.623
     128   8.85  95.67%   0.92%   3.42%   2511.276       2.511
     256  15.95  95.69%   0.82%   3.49%   2788.855       2.789

Memory required:  15645K.


LINPACK benchmark
long double precision
Digits:  18
Array size 1000 X 1000.
Average rolled and unrolled performance:

    Reps Time(s) DGEFA   DGESL  OVERHEAD    MFLOPS      GFLOPS
----------------------------------------------------------------
       2   0.58  97.73%   0.68%   1.60%    589.918       0.590
       4   1.23  97.96%   0.60%   1.45%    554.509       0.555
       8   2.47  97.89%   0.61%   1.51%    552.373       0.552
      16   4.69  97.87%   0.62%   1.51%    580.605       0.581
      32   9.25  97.89%   0.61%   1.49%    589.104       0.589
      64  18.97  97.88%   0.62%   1.50%    574.174       0.574

Memory required:  15645K.


LINPACK benchmark
__float128 precision
Digits:  33
Array size 1000 X 1000.
```

```
Average rolled and unrolled performance:

    Reps Time(s) DGEFA   DGESL  OVERHEAD    MFLOPS      GFLOPS
----------------------------------------------------------------
       1   5.41  98.99%   0.54%   0.47%      31.157       0.031
       2  10.05  98.97%   0.57%   0.45%      33.509       0.034
```

Problem 5 [10 points] I will comment in class on this one.

Problem 6 [6 points] Compute an accurate sum using `vpa` See the code.