V63.0252 Numerical Analysis Spring 2007
January 24, 2007

Professor Olof Widlund
Office: CIWW 712, 251 Mercer Street
Phone: 212 998-3110
Electronic mail: widlund@cims.nyu.edu
Course home page URL: http://www.math.nyu.edu/courses/spring07/V63.0252-001/index.html
Office hours: Mondays 3:30–4:30pm and Thursdays 4:00–5:00pm.

Homework set 1: Due Monday February 5, at midnight.
No homework will be accepted after that time.

Homework should be given to me in class or put under my office door. Do not put it in my mail box. For general rules, read my home page.

If you do not have previous experience with Matlab, find time to start using it. Note that Matlab uses double precision IEEE arithmetic. Make sure to use format long.

Did anyone find a good free introduction to Matlab programming on the web? If so, please share it with the class and me; I could send out an e-mail message to all students with the information.

1. What is the smallest positive integer which is not exactly represented as a single precision IEEE floating point number? What is the largest finite integer which is part of the double precision IEEE floating point system?

2. Find the IEEE single and double precision floating point representation of the numbers $4, 100, 1/100, 2^{-100}, 2^{-200}$, and 2^{-1050}. Assume that we use rounding to nearest.

3. Find, by using Matlab, the first power of 10 that is not exactly represented in the IEEE double precision system.

   What about powers of 10^{-1}?

   Can you explain your answers by looking in the table of double precision numbers given in the recent handout?

4. Find three floating point numbers a, b, c, such that $a+b+c$ is computed with a quite large relative error. Try it out using Matlab.

5. Consider the following matlab program:

```
f(1)=1;
f(2)=1;

for i=2:(n-1)
f(i+1) = f(i) + f(i-1);
end

g(n)=f(n);
g(n-1)=f(n-1);
for i=(n-1):-1:2
g(i-1)=g(i+1) - g(i);
end
```

For any input n the program computes the n first Fibonacci numbers in the normal order and it then computes the same sequence of numbers in reverse order. For small enough n there is no round-off errors at all. Explain why. For very large values of n, the numbers will overflow. Find out how large n must be for this to happen. Finally, for values of n in between, the $f(i)$ will differ very much from the $g(i)$. Get some experience with this and provide as full an explanation as you can for what is happening. Discuss your finding in light of what is known about the IEEE double precision system.