Numerical Analysis **Answers to Assignment #3**

1. Differentiating $[\xi(a)]^n - ae^{-\xi(a)} \equiv 0$ with respect to a gives

$$n[\xi(a)]^{n-1}\xi'(a) - e^{-\xi(a)} - ae^{-\xi(a)}(-\xi'(a)) = 0,$$
$$\xi'(a)(ae^{-\xi(a)} + n[\xi(a)]^{n-1}) = e^{-\xi(a)},$$

hence

$$\xi'(a) = \frac{e^{-\xi(a)}}{ae^{-\xi(a)} + n[\xi(a)]^{n-1}} = \frac{e^{-\xi(a)}}{[\xi(a)]^n + n[\xi(a)]^{n-1}} = \frac{e^{-\xi(a)}}{[\xi(a)]^{n-1}(\xi(a)+n)} .$$

Therefore,

$$(\text{cond } \xi)(a) = \left|\frac{a\xi'(a)}{\xi(a)}\right| = \frac{ae^{-\xi(a)}}{[\xi(a)]^n(\xi(a)+n)} = \frac{1}{\xi(a)+n} .$$

Since $\xi(a) > 0$, one clearly has

$$(\text{cond } \xi)(a) < \frac{1}{n}.$$

2. By the law of machine arithmetic,

$$y_A = \text{fl}(x_1 + x_2) = (x_1 + x_2)(1+\varepsilon), \quad |\varepsilon| \le \text{eps}.$$

Hence, we can write

$$y_A = x_1(1+\varepsilon) + x_2(1+\varepsilon) = f(x_A), \quad x_A = [x_1(1+\varepsilon),\ x_2(1+\varepsilon)]^T.$$

There follows

$$\frac{\| x - x_A \|_1}{\| x \|_1} = \frac{|x_1\varepsilon| + |x_2\varepsilon|}{|x_1| + |x_2|} \le \frac{|x_1| + |x_2|}{|x_1| + |x_2|} \text{ eps} = \text{eps},$$

which means that

$$\gamma(x_1, x_2) = (\text{cond}\, A)(x) \le 1.$$

The same is obtained for the other norms. The algorithm A is thus perfectly well conditioned. Nevertheless, the result y_A can be inaccurate if f is ill conditioned (cf. Problem 25(d) or 28):

$$\left|\frac{y_A - y}{y}\right| = \left|\frac{f(x_A) - f(x)}{f(x)}\right| \approx (\text{cond}\, f)(x)\ \frac{\| x_A - x \|}{\| x \|}\ \le\ \kappa(x_1, x_2) \text{ eps}.$$

The inaccuracy, however, must be blamed on f, not on the algorithm A.

3. PROGRAM

```
      double precision dgamma,dsum,dk,dgam,derr,das
      dgamma=.57721566490153d0
      dsum=0.d0
      do 10 k=1,10000
        dk=dble(k)
        dsum=dsum+1.d0/dk
        if(1000*(k/1000).ne.k) goto 10
        dgam=dsum-dlog(dk)
        derr=dgamma-dgam
        das=dk*derr
        write(*,2) k,das
 2      format(i10,d20.7)
10    continue
      stop
      end
```

RESULTS

```
     k       k(gamma-gamma_k)

   1000       -0.4999167D+00
   2000       -0.4999583D+00
   3000       -0.4999722D+00
   4000       -0.4999792D+00
   5000       -0.4999833D+00
   6000       -0.4999861D+00
   7000       -0.4999881D+00
   8000       -0.4999896D+00
   9000       -0.4999907D+00
  10000       -0.4999917D+00
```

The results strongly suggest that $d = 1$ and $c = -\frac{1}{2}$. A verification can be had from Eqs. 6.3.2 and 6.3.18 in Abramowitz & Stegun.

4. PROGRAM

```
      double precision dsum,dk
      dsum=0.d0
      do 10 k=1,1000000
        dk=dble(float(k))
        dsum=dsum+1.d0/((1.d0+dk)*dsqrt(dk))
        if(100000*(k/100000).eq.k) write(*,2) k,dsum
 2      format(1x,i18,d15.7)
10    continue
      stop
      end
```

RESULTS

k	s_k
100000	.1853701D+01
200000	.1855553D+01
300000	.1856374D+01
400000	.1856863D+01
500000	.1857197D+01
600000	.1857443D+01
700000	.1857635D+01
800000	.1857789D+01
900000	.1857917D+01
1000000	.1858025D+01

The exact value of the sum is

$$s = 1.86002507922\ldots$$

Evidently, the series converges extremely slowly.

5. For n sufficiently large, $1 + 1/n$ will become equal to 1 because of rounding. From this point on, the "machine limit" will be 1 instead of e. This is borne out by the following computer run.

PROGRAM

```
   do 10 in=1,14
     fn=10.**in
     e=(1.+1./fn)**fn
     write(*,2) fn,e
 2   format(1x,e10.2,e15.7)
10 continue
   stop
   end
```

RESULTS

```
    n      fl((1+1/n)^n)

 .10E+02   .2593743E+01
 .10E+03   .2704811E+01
 .10E+04   .2717051E+01
 .10E+05   .2718597E+01
 .10E+06   .2721962E+01
 .10E+07   .2595227E+01
 .10E+08   .3293968E+01
 .10E+09   .1000000E+01
 .10E+10   .1000000E+01
 .10E+11   .1000000E+01
 .10E+12   .1000000E+01
 .10E+13   .1000000E+01
 .10E+14   .1000000E+01
 .10E+15   .1000000E+01
```