Sample Computer Program Write-up
Fall, 2000

The assignment was to write a program to do the bisection method for the function

$$f(x) = e^x - 2,$$

using the initial interval $[0, 1]$. The program was written in FORTRAN, using the pseudo-code in the text as a guide. The FORTRAN code is attached.

The table below summarizes the results for a desired accuracy of 10^{-4}. The approximate root is $c_{14} = 0.69317627$. Note that the exact root is $\ln 2 = 0.6931471806$, so that our error is

$$|\ln 2 - c_{14}| = 2.90894 \times 10^{-5} \leq 10^{-4}$$

as required.

Table 1: Bisection method applied to $f(x) = e^x - 2$.

n	a_n	c_n	b_n	$f(c_n)$
1	0.50000000	0.50000000	1.00000000	-0.35127878
2	0.50000000	0.75000000	0.75000000	0.11700010
3	0.62500000	0.62500000	0.75000000	-0.13175404
4	0.68750000	0.68750000	0.75000000	-0.01126254
5	0.68750000	0.71875000	0.71875000	0.05186677
6	0.68750000	0.70312500	0.70312500	0.02005553
7	0.68750000	0.69531250	0.69531250	0.00433540
8	0.69140625	0.69140625	0.69531250	-0.00347888
9	0.69140625	0.69335938	0.69335938	0.00042439
10	0.69238281	0.69238281	0.69335938	-0.00152814
11	0.69287109	0.69287109	0.69335938	-0.00055206
12	0.69311523	0.69311523	0.69335938	-0.00006390
13	0.69311523	0.69323730	0.69323730	0.00018024
14	0.69311523	0.69317627	0.69317627	0.00005817

Program listing:

```
c
c       Code for sample bisection algorithm
c
        f(x) = exp(x) - 2.0
c
        tol = 1.e-4
        a = 0.0
        b = 1.0
        fa = f(a)
        fb = f(b)
c
        zn = (alog(b-a) - alog(tol))/alog(2.0)
        n = ifix(zn) + 1
c
        do k=1,n
                c = a + 0.5*(b - a)
                fc = f(c)
                if(fa*fc .lt. 0.0) then
                        b = c
                        fb = fc
                else
                        a = c
                        fa = fc
                endif
                write(6,10) k,a,c,b,fc
10              format(1h ,i5,4(' & ',f12.8) '\\\\')
                if(fc .eq. 0.0) then
                        print*,'Premature convergence!'
                        stop
                endif
        enddo
c
        stop
        end
```

Raw output:

```
 1 &   0.50000000 &   0.50000000 &   1.00000000 &  -0.35127878\\
 2 &   0.50000000 &   0.75000000 &   0.75000000 &   0.11700010\\
 3 &   0.62500000 &   0.62500000 &   0.75000000 &  -0.13175404\\
 4 &   0.68750000 &   0.68750000 &   0.75000000 &  -0.01126254\\
 5 &   0.68750000 &   0.71875000 &   0.71875000 &   0.05186677\\
 6 &   0.68750000 &   0.70312500 &   0.70312500 &   0.02005553\\
 7 &   0.68750000 &   0.69531250 &   0.69531250 &   0.00433540\\
 8 &   0.69140625 &   0.69140625 &   0.69531250 &  -0.00347888\\
 9 &   0.69140625 &   0.69335938 &   0.69335938 &   0.00042439\\
10 &   0.69238281 &   0.69238281 &   0.69335938 &  -0.00152814\\
11 &   0.69287109 &   0.69287109 &   0.69335938 &  -0.00055206\\
12 &   0.69311523 &   0.69311523 &   0.69335938 &  -0.00006390\\
13 &   0.69311523 &   0.69323730 &   0.69323730 &   0.00018024\\
14 &   0.69311523 &   0.69317627 &   0.69317627 &   0.00005817\\
```