

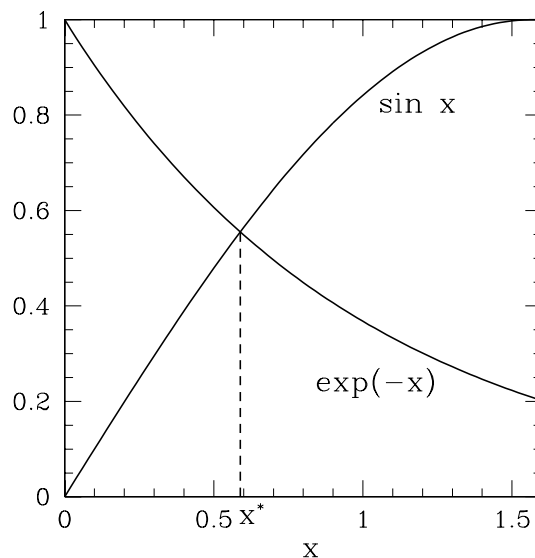
Physics 115/242; Peter Young
Some output from root finding algorithms.

We consider the problem of finding the root of

$$\boxed{f(x) \equiv e^{-x} - \sin x = 0} \quad (1)$$

for x in the interval from 0 to $\pi/2$. This corresponds to the intersection of the curves e^{-x} and $\sin x$ shown in the figure. We denote the root by x^* . It's value is is

$$\boxed{x^* = 0.5885327439818611\dots} \quad (2)$$



I. BISECTION METHOD

We take the desired accuracy to be 10^{-4} , and use the starting values $x_l = 0.4, x_g = 0.8$. Successive estimates are then

n	xl	xg
1	0.40000	0.60000
2	0.50000	0.60000
3	0.55000	0.60000
4	0.57500	0.60000
5	0.58750	0.60000
6	0.58750	0.59375
7	0.58750	0.59063
8	0.58750	0.58906
9	0.58828	0.58906
10	0.58828	0.58867
11	0.58848	0.58867
12	0.58848	0.58857

The program stopped when $|x_g - x_l| < 10^{-4}$ and gives

$$\boxed{x = 0.5885}, \quad (3)$$

which is correct to 4 decimal places. The uncertainty in the value of x^* after n interactions, $\epsilon_n \equiv x_g - x_l$, varies as

$$\boxed{|\epsilon_n| = \frac{1}{2} |\epsilon_{n-1}|}, \quad (4)$$

so the number of decimal places of accuracy increases proportional to n , which we call (in this context) *linear convergence*.

II. SECANT METHOD

We take the desired accuracy to be 10^{-10} and use take starting values $x_0 = 0.4$ and $x_1 = 0.8$. Subsequent values are:

n	x_n	x_n - x_{n-1}	x_n - exact	bracketed?
2	0.604690800056	0.195309200E+00	0.161580561E-01	not bracketed
3	0.586997111757	0.176936883E-01	-0.153563222E-02	bracketed
4	0.588542746639	0.154563488E-02	0.100026570E-04	bracketed
5	0.588532750126	0.999651316E-05	0.614386464E-08	not bracketed
6	0.588532743982	0.614388918E-08	-0.245359288E-13	bracketed
7	0.588532743982	0.245359288E-13	0.000000000E+00	

The program stopped when $|x_n - x_{n-1}| < 10^{-10}$, and gives

$$\boxed{x = 0.5885327440}, \quad (5)$$

correct to 10 decimal places. The fourth column, which gives the error in x_n shows that the final value of x actually agrees with the correct value to machine precision (about 10^{-16}), much more accurate than the precision specified. The last column indicates whether the last two values for x_n bracket the root or not. There is apparently no simple pattern to this.

In class, we stated without proof that the number of decimal places of accuracy typically increases by a factor of the golden ratio, $1.618 \dots$ on each iteration, i.e., if ϵ_{n-1} is small,

$$\boxed{|\epsilon_n| = C |\epsilon_{n-1}|^{1.618}}, \quad (6)$$

where C is a constant and ϵ_n is the error in x_n . For a derivation see <http://www.math.uic.edu/~leykin/mcs471/NOTES/secant.pdf> and

<http://www.mathpath.org/Algor/squareroot/secant.pdf>.

The data is roughly consistent with this, *faster than linear*, convergence. Note, however, that the root does not remain bounded by the last two values of x_n , and convergence is not guaranteed if the initial values, x_0 and x_1 , are far from the root even if they bound it.

III. NEWTON-RAPHSON METHOD

We take the desired accuracy to be 10^{-14} and the starting values are $x_0 = 0.8$. Subsequent values are

n	x_n	$ x_n - x_{n-1} $
1	0.5661267157835728	0.233873284216E+00
2	0.5883360593267145	0.222093435431E-01
3	0.5885327285001999	0.196669173485E-03
4	0.5885327439818610	0.154816610642E-07
5	0.5885327439818611	0.111022302463E-15

The program stopped when $|x_n - x_{n-1}| < 10^{-14}$, and gives

$$\boxed{x = 0.58853274398186,} \quad (7)$$

correct to 14 decimal places.

Note that the data is consistent with the expected result that the number of decimal places of accuracy doubles on each iteration,

$$\boxed{|\epsilon_n| = C |\epsilon_{n-1}|^2,} \quad (8)$$

if ϵ_{n-1} is small. In other words we have *quadratic convergence*, which is very rapid. On the other hand, depending on the problem and the starting value of x , convergence may not occur at all!