QUICK AND DIRTY FUNCTIONS FOR BASIC

BY DENNIS ALLISON
169 Spruce Ave.
Menlo Park, CA 94025

Recently, Gene Wallis called my attention to the fact that a number of BASIC interpreters for the hobbyist are not provided with a full complement of intrinsic functions. He was working on an energy utilization program and needed sine/cosine, log, and power functions badly. These programs are quick and dirty implementations which I did to give him a tool to get his job done.

A word of warning though. These are *quick and dirty* in the sense that they have not been carefully tuned or tested. They may well contain errors. Certainly the choice of approximation may not be optimal. All the functions are computed by using various range reducing formulae and then applying a polynomial or rational approximation from Hart's *Computer Approximations* (Wiley, 1968), an encyclopedic compendium of approximations which is now out of print. The approximations given in the algorithm section are all accurate to better than seven digits, but the programs should be trusted to only six digits or so because of truncation of the coefficients, carelessness of implementation, and the vagaries of local floating point arithmetic.

The programs are also given in BASIC. They have been tested using the dialect of BASIC supported by the Unix time-sharing system. The Unix BASIC differs substantially from most hobby BASICS, so the test programs were transliterated into something more generally acceptable. Beware—errors may have crept into the code.

Lastly, if you do use the programs, considerable efficiency can be gained by tuning the code to your particular BASIC. In writing the functions we chose the simplest possible code rather than the most efficient. A little rewriting may speed the functions enormously. All these functions are to be invoked via gosubs; they expect an argument in x and return a value in y. $x0, x1, x2, \ldots$ are used as temporaries.

Square Root—SQR(x)

1. $x < 0$ is an error
2. reduce range
   $t = 1$
   $\{x = x/100; t = t * 10\}$ until $x \leqslant 1$
   note: $x * t * t =$ original number
   $0 \leqslant x \leqslant 1$
3. make a guess at $\sqrt{x}$ via a polynomial (accuracy is .56 digits):
   $y_0 = .115442 + 1.15442 * x$
4. iterate until $|y_{n+1} - y_n| < 10^{-6}$ (or so—depends upon floating point accuracy) using:
   $$y_{n+1} = ½\left(y_n + \frac{x}{y_n}\right) = y_n + ½\left(\frac{x}{y_n} - y_n\right)$$
5. final answer is:
   $y_{n+1} * t$

You don't really have to do the test in 4. as the iteration will double the accuracy (approx) every time. So:

	iteration	accuracy (digits)
	0	0.5
	1	1
	2	2
	3	4
should be enough →	4	8
certainly enough →	5	16

```
500 rem y = sqrt(x)
510 if x<0 then goto <error>
520 x1 = 1
530 if x <=1 then goto 570
540 x = x/100
550 x1 = x1 * 10
560 goto 530
570 y = .115442 + 1.15442*x
580 x2 = 5
590 y = 0.5*(y+(x/y))
600 x2 = x2 - 1
610 if x2>0 then goto 590
620 y = y*x1
630 return
```

Log Base 10

1. $x \leqslant 0$ is an error
2. $x < 1$ then $\left\{ \text{sign} = -1 \quad x = \dfrac{1}{x} \right\}$ else sign = 1
3. reduce range. find n such that

   since $\log\left(\dfrac{1}{x}\right) = -\log(x)$

   $$\frac{1}{10} \leqslant \frac{x}{10^n} < 1$$

   and set $x = \dfrac{x}{10^n}$

4. $s = \sqrt{10}\; x$
5. $z = (s - 1) \;/\; (s + 1)$

   $$r = z\,\frac{P(z^2)}{Q(z^2)}$$

6. $\text{lgt} = (n + r - .5) * \text{sign}$

P_0	3.1878	22082	024
P_1	−2.6558	08794	66
P_2	.26686	32700	47
Q_0	3.6701	15625	115
Q_1	−4.2809	73292	83
Q_2	1.0		

```
500 rem y = log(x)  base 10
510 if x <= 0 then goto <error>
520 x1 = 1
530 if x < 1 then x1 = -1
540 if x < 1 then x = 1/x
550 x2 = 0
560 if x < 1 then goto 600
570 x = x/10
580 x2 = x2 + 1
590 goto 560
600 x = 3.162278*x
610 x = (x+1)/(x-1)
620 y = x*x
630 x3=3.187822+y*(-2.655809+y*0.2668633)
640 x4=3.670116+y*(-4.280973+y)
650 y = (x2 + x*x3/x4 -.5)*x1
660 return
```

Power 10^x

1. alarms if x too large positively
   return zero if large negatively
2. if $x < 0$ $\{ \text{inv} = \underline{\text{true}},\; x = |x| \}$
   else $\{ \text{inv} = \underline{\text{false}} \}$
   reduce range to $[0, ½]$

   $n = \text{int}(x)$

   $x = x - n - .5$

   $$z = \frac{2x\,P(x^2)}{Q(x^2) - x\,P(x^2)}$$

result = $\sqrt{10}\; 10^n\,(z + 1)$
if (inv) $\Rightarrow$ result = 1 / result

P_0	69.5569663167251
P_1	8.7517567967368
Q_0	60.4164133787158
Q_1	34.2951432306918
Q_2	1

```
500 rem y = 10^x
510 x1 = 0
520 if x < 0 then x1 = 1
530 if x < 0 then x = -x
540 x2 = int(x)
550 x = x - n - .5
560 y = x * x
570 x3 = 69.55697+y*8.751758
580 x4 = 60.41641+y*(34.29514+y)
590 y = (2*y*x3)/(x4-y*x3)
600 if x2 = 0 then goto 640
610 y = y * 10
620 x2 = x2 - 1
630 goto 600
640 if x1 = 1 then y = 1/y
650 return
```

Inverse Trig

1. If $x < 0$ return $-\text{atan}(|x|)$
   since $\text{atan}(-x) = -\text{atan}(x)$

2. If $x > 1$ return $\frac{\pi}{2} - \text{atan}(\frac{1}{x})$

   since $\text{atan}(|x|) = \frac{\pi}{2} - \text{atan}(\frac{1}{|x|})$ and $x > 0$

3. compute $\text{atan}(x)$ in $[0, \tan\frac{\pi}{4} = 1]$
   by:

$$z = x * x$$

$$\text{atan}(x) = x \frac{P_0 + P_1 z + P_2 z^2}{Q_0 + Q_1 z + Q_2 z^2 + Q_3 z^3}$$

P_0	80.78998484078
P_1	72.5814378046
P_2	11.11774163116
Q_0	80.78998647615
Q_1	99.5112677199
Q_2	28.13285868067
Q_3	1.0

```
500 rem y = atan(x)
510 x1 = 1
520 if x < 0 then x1 = -1
530 if x < 0 then x = -x
540 x2 = 0
550 if x > 1 then x2 = 1
560 if x > 1 then x = 1/x
570 y = x*x
580 x3=80.78998+y*(72.58144+y*11.11774)
590 x4=80.78999+y*(99.51128+y*(28.13286+y))
600 y = y*x3/x4
610 if x2 = 1 then y = 1.5707963 - y
620 y = y*x1
630 return
```

Trig Functions

Sine: Compute as cosine using:

$$\sin(x) = \cos(\frac{\pi}{2} - x)$$

Cosine: 1. test arg too large
$|x| > 10^{d-3}$, $d =$ # of digits in floating point
then error

2. $x = |x|$ since $\cos(-x) = \cos(x)$
3. $x = \text{mod}(x, 2\pi)$ since
   $\cos(2\pi + x) = \cos x$. Range is now
   $0 \leqslant x < 2\pi$
4. $0 \leqslant x < \frac{\pi}{2}$: $s = +1$ $\quad x = x$

   $\frac{\pi}{2} \leqslant x < \pi$: $s = -1$ $\quad x = |\pi - x|$

   $\pi \leqslant x < \frac{3\pi}{2}$: $s = -1$ $\quad x = |\pi - x|$

   $\frac{3\pi}{2} \leqslant x < 2\pi$: $s = +1$ $\quad x = |x - 2\pi|$

5. $z = x^2$
   $y = P_0 + P_1 z + P_2 z^2 + P_3 z^3 + P_4 z^4$
   $\cos = y * s$.

P_0	.999999953464
P_1	−.499999053455
P_2	.0416635846769
P_3	−.0013853704264
P_4	.00002315393167

```
500 rem y = sin(x)
510 x = 1.570796 - x

520 rem cos(x)
530 if x < 0 then x = -x
540 if x > 1000 then goto <error>
550 x = x - int(x/6.283185)*6.283185
560 x2 = int(x/1.5707963)
570 x2 = 1
580 if x1 = 1 then x2 = -1
590 if x1 = 2 then x2 = -1
600 if x1 = 1 then x = 3.141593 - x
610 if x1 = 2 then x = 3.141593 - x
620 if x1 = 3 then x = x - 6.283185
630 x1 = x*x
640 y=0.9999999+x1*(-.4999991+x1*(.04166359+x1*(
    -0.001385370+x1*(0.00002315393))))
650 y = y * x2
660 return
```