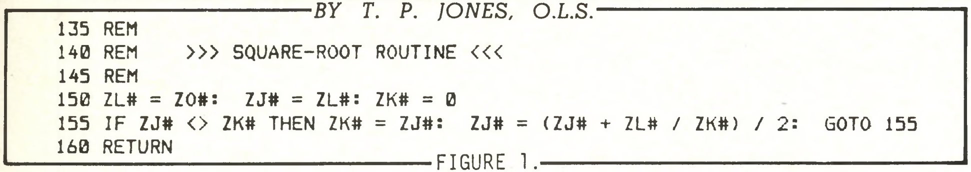
Basic Mathematics

BY T. P. JONES, O.L.S.



```
135 REM
140 REM    >>> SQUARE-ROOT ROUTINE <<<
145 REM
150 ZL# = ZO#:  ZJ# = ZL#: ZK# = 0
155 IF ZJ# <> ZK# THEN ZK# = ZJ#:  ZJ# = (ZJ# + ZL# / ZK#) / 2:  GOTO 155
160 RETURN
```

FIGURE 1.

NUMERIC FUNCTIONS included within the BASIC programming language will generate a sin, cosine, tangent, arc-tangent, square root, etc. etc., of an argument any time a particular function is called.

A significant disadvantage of the functions, however, is that the result presented is nearly always to single precision. That is, to only six or seven significant digits.

As we have traditionally used seven figure tables, and as these tables are absolutely necessary in some cases, (e.g. a calculation of a curve intersection having a large radius), single precision is not really good enough for the work that we have to do.

But because BASIC is such an easy-to-use language, and is so powerful for interactive use, especially during program development and debugging, some routines have been written to compute trigonometric functions, square roots, etc., to sixteen significant digits (double precision).

It may be that this work has already been done and published. In fact, it seems a little inconceivable that it has not. If anyone is aware of similar material, they would confer a favour by so advising the Editor. There is not much point in taking up space in this journal with data that may already be available elsewhere.

For those readers who wish to write survey programs for their own micro-computers, these sub-routines will be presented here during the next few editions of the Quarterly. Non-members of our Association may wish to bring their subscriptions up-to-date, so as to be sure that none of the routines will be missed!

NOTE

While the routine listed here has undergone extensive testing to be sure that it operates properly, neither myself or the Editorial Board make an express or implied warranty of any kind with regard to it. In no event will we, or The Association of Ontario Land Surveyors, be liable for incidental or consequential damages in connection with or arising out of the furnishing, performance or use of any of these programs.

All variables, or work spaces, within the sub-routines are identified by two alpha characters, the first of which is always Z. The #, and % signs force the variables to be double precision, or integers, respectively. Depending upon the vagaries of the 'BASIC' language being used, operators may decide to use a "DEFDBL", or some such command, to classify the variables as double precision.

ZA # to ZO # inclusive will be double precision, and ZP% to ZU% will be integers. In all of the routines, the argument will be ZO #, and the result will be ZJ%.

The first routine, "Square Root", is listed in Figure 1. A simple program to access and test it, could be as follows:-

```
10  INPUT ZO # :  GOSUB 40075
20  PRINT ZJ # :  GOTO 10
```

Just key RUN to make it go. ●