```
TEN:
     AAP=2.0/(9.0*AP);
     BBP=2.0/(9.0*BP);
IF(BBP=0) | (FP=0) | (AAP< 0) THEN DO;
     PUT DATA(BBP,FP,AAP);
     PUT LIST('SQRT OF 0');
GO TO FERN;
END;
     ZP=ABS(((1.0-BBP)*FP**0.333333-1.0+AAP)/SQRT(BBP*FP**0.666667+
     AAP));
FERN:
     IF BP< 4 THEN ZP=ZP*(1.0+.8*ZP**4/BP**3);
     PROB=0.5/(1.0+ZP*(.196854+ZP*(.115194+ZP*(.000344+ZP*
     .019527))))**4;
IF FR< 1.0 THEN PROB=1.0-PROB;
     RETURN(PROB);
END PRBF;
END;
```

REFERENCES

BRUNING, J. L., & KINTZ, B. L. *Computational handbook of statistics.* Glenview, Ill: Scott, Foresman, 1968. Pp. 61-72.

LINDQUIST, E. F. *Design and analysis of experiments in psychology and education.* Boston: Houghton Mifflin, 1953. Pp. 281-284.

LIPPMAN, L. G., GALOSY, R. A., & THOMPSON, R. W. Passive avoidance learning in gerbils and rats. Journal of Comparative & Physiological Psychology, 1970, 73, 269-273.

VELDMAN, D. J. *Fortran programming for the behavioral sciences.* New York: Holt, Rinehart & Winston, 1967.

High-speed point plotting with computer-controlled displays[*]

L. WOJNAROWSKI, K. BACHMAN, and I. POLLACK[†]
University of Michigan, Ann Arbor, Michigan 41804

A method is described for the high-speed plotting of many independently generated elements in a computer-controlled display.

We shall describe a method of high-speed point plotting which we have found to be extremely useful. We are aware that special-purpose hardware-software display systems, e.g., character generators, vector plotters, are available to simplify certain display programs and to gain speed. The present method may be regarded as a supplement to, rather than a replacement of, such special-purpose display systems. The method may be expected to be useful when an extremely large number of independently generated points must be plotted within a short period of time.

The specific problem we faced was to paint four 32 x 32 element binary-coded displays upon a high-speed display scope (Tektronics 602 with P15 phosphor) without noticeable flicker. The two binary states—dot and no-dot—were equiprobable. We thus sought to plot the positions of approximately 0.5 x 4 x 32 x 32, or 2,048, independently calculated dots. In order to avoid flicker upon successive paintings, we aimed to paint succcessive displays within 25, and preferably within 20, msec.

Several schemes are obvious candidates for consideration. Each may be the solution for a particular display system, depending upon the tradeoffs available for speed and for storage.

Scheme A is the most economical of machine storage, but is the slowest for displays with a large number of points. Displays are painted point by point as each point is calculated. No intermediate storage is required. This method of "on-line" plotting is excellent for displays with a very small number of display elements. The method is, however, hampered by the intermediate computational time required, especially for displays with a large number of display elements.

Schemes B and C are more costly of machine storage and are slower than Scheme A in terms of the delay from the beginning of computation to the display of the first point. The display cycle, however, is faster than for Scheme A, especially with a large number of display elements. Instead of computing and painting individual dots as in Scheme A, Schemes B and C first compute the binary state of each point and store the results of the successive computations. If the same display is presented over and over for successive paintings, the computation need be performed only once. In its most economical form for storage, Scheme B stores the binary code. The plotting routine scans the stored binary code, and in a plotting routine with a fixed geographical sequence, plots each binary element "1" and does not plot each binary element "0." This system is relatively economical of storage since the binary coding is stored, and the individual words can be "unpacked" bit by bit. In our laboratories, Lon Radin has successfully used this method to plot four 60 x 60 element matrices with the PDP-8I computer (4,096 12-bit words). A storage display scope must be employed, however, because the conversion from binary code to geographical position takes too long for flickerless displays.

Scheme C, like Scheme B, first performs all of the necessary computations to determine the binary state of each display element. Scheme C carries the computations one step further. Instead of storing the binary code, Scheme C computes and stores the coordinates of each to-be-plotted point. Scheme C is more wasteful of storage than Scheme B, since the coordinate, rather than the binary state, information must be stored. On the other hand, because only the coordinates of the to-be-plotted locations are stored, the no-dot elements may be ignored. The resulting display routine can be made extremely tight.

In our first use of Scheme C, we picked up the x coordinate of the to-be-plotted point and instructed the x digital-to-analog converter to generate and hold the respective x voltage; we then picked up the y coordinate of the to-be-plotted point and instructed the y digital-to-analog converter to generate and hold the respective y voltage. Then we fired the z axis of the display scope. The steps were: fetch x, generate and hold x, fetch y, generate and hold y, fire z. The display cycle alone took 16 microsec, and, with housekeeping, took 22 microsec per point on the PDP-9 computer. The result was unsatisfactorily long—2,048 x 22 microsec, or 44 msec—for the displays we planned to use.

We had exhausted the three schemes without solving our initial goal and became discouraged. Then we got lucky. We noted that we were not using the full 12 bits of our digital-to-analog converters. We figured that we could get away with only 9 bits' resolution upon each coordinate, since plotting took place upon only 64 lines. We decided to store both the x and y coordinates of each point within a single word of the

[*] The activities were supported, in part, by Grant 14036X of the National Science Foundation.

[†] Mailing address: Irwin Pollack, Mental Health Research Institute, 205 N. Forest Street, Ann Arbor, Michigan 48104.

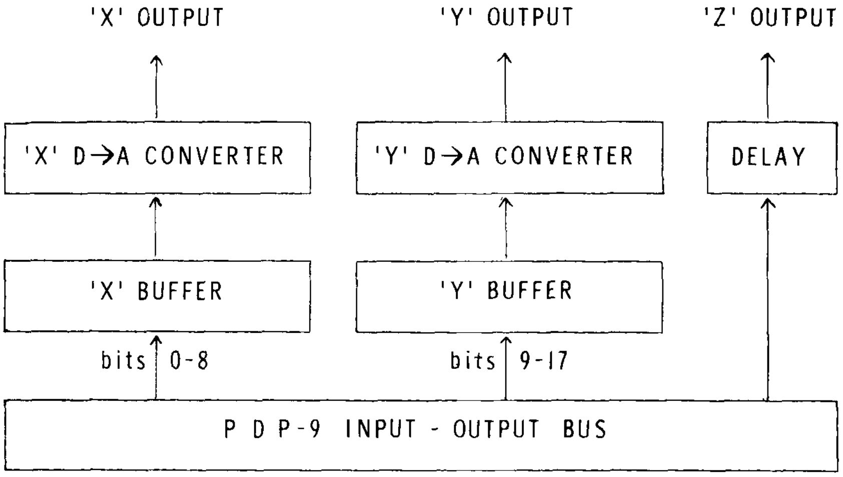

Fig. 1. Schematic implementation of high-speed point plotting.

PDP-9 computer (8,192 18-bit words). The desired sequence, illustrated in Fig. 1, then became clear. The x and y coordinates are placed into the input-output bus of the PDP-9. In one operation, we fetch both coordinates and stuff both coordinates into the 18-bit input-output buffer. The x D-to-A converter is hooked to Bits 0-8 of the input-output buffer; the y converter is hooked to Bits 9-17. There is no requirement to store and hold either voltage before the z axis of the display score is fired, except for a fixed delay required to allow the converters to settle down before their outputs are displayed. The steps then were: fetch xy, sample xy, and fire z. The display cycle was reduced to 6 microsec, and, with housekeeping, takes only 9 microsec for each plotted point. For four 32 x 32 element displays upon a single painting, the plotting time is 18 msec (2,048 x 9 microsec) which met our goal.

We do not propose the present method as the solution for every display problem. Merely, we say that the scheme solved our problem, and it may solve yours. We were lucky in that our time requirements permitted computation time before display and that our resolution requirements permitted the simultaneous storage of both x and y coordinates within a single word. We were willing to trade off the computational simplicity of Scheme A and the more economical storage of Scheme B for the extra display speed of Scheme C. Even Scheme C would have been unsuccessful had we been unable to fire both coordinates simultaneously.

The problem of display speed can be avoided by the use of a storage display scope. However, the lack of precisely controlled display times and the long persistence of most storage scopes may be unacceptable to some display applications. We have been able to squeeze out four 48 x 48 element displays within a single painting on the PDP-9, but flicker or ripple is encountered with the high-speed phosphor. The resulting display time is 41.5 msec, or 4,608 x 9 microsec. Thus, even Scheme C may be inadequate with displays of still larger number of elements than we employed. Although the high-speed display feature is limited to one-half of the bits of the accumulator, the lower ordered bits on the converter can be set independently, but not at high speed. For example, with a 10-bit D-to-A converter, assume that a display is generated by controlling upon the high-ordered 9 bits of the converter, with the lowest ordered bit set to 0. The next display can employ the lowest ordered bit set to 1. In this way, for example, slight displacements can be introduced which draw upon the full power of the converter.