

It can similarly be shown that the grand total of K members from the start of the distribution and $R-K$ members from the end of the distribution obeys the same constraint.

Yours faithfully

R. G. ELSTON
South West Thames Regional Health
Authority
Regional Computer Centre
Springfield Hospital
61 Glenburnie Road
London SW17 7DJ

Reference

1. A. J. Compton, An algorithm for the even distribution of entities in one dimension. *The Computer Journal* **28** (5), 530–537 (1985).

An Algorithm for the Even Distribution of Entities in one Dimension

Author's comments on letters received by The Computer Journal from K. Brokate and R. G. Elston

I am interested to note that both correspondents use similar line-plotting problems as examples requiring evenness.

R. G. Elston's method is similar to delta-estimation in control, where the next direction of a control signal depends on which side of the ideal the controlled variable is at that stage. For any system with a first-order response and thus no overshoot, this method is certainly simpler than mine. I appreciate the proof and the concise definition of perfect evenness he includes.

The problem I originally addressed and from which the ideas in the paper emerged concerned the thermal control of a small, radiating, highly non-linear system using burst-firing thyristors. I needed to calculate the size of the next energy impulse from several previous readings, then evenly distribute a block of heating pulses within the next time slot. A simple delta-estimation method would have caused large oscillations. This also explains, to answer K. Brokate, why I was not concerned with symmetry; as I mentioned in section 4, I regarded 101101 as equivalent to 110110 as the sequences were to be repeated anyway. Symmetry is obviously important in the line-drawing examples.

Turning to my definition using the D.F.T., there are several approximate but quick

methods for evening out pulses – bit reversal and binary rate multiplication are two – and I wanted a definition of evenness that indicated quality, not just an absolute 'even' or 'not even', to compare these methods.

As far as the algorithm itself is concerned, I developed it partly because of its intrinsic interest in defining an evenly spread series from just a few numbers (ps and qs in the article). It has a practical use, however, in sharing the calculation time between that before a run and that during it and being economical of storage space. Elston's method could presumably be used either (i) before a run, filling an array with the required directions for it or (ii) during the run, making the corrections as required. The software overheads for these and my method are shown in Fig 1:

K. Brokate's algorithm may well be as good as or better than mine, but would need to be expressed as a program fragment to enable a comparison to be made. I make no claim for uniqueness in my algorithm, and was aware when I wrote the article that I might have been re-inventing the wheel – but there are many kinds of wheel, each with a different purpose.

A. J. COMPTON

Method	Calculations done before run		Time for calculations done during run	Comments
	Time	Storage space		
(a) Elston (i)	Large	Large (all 0s and 1s)	Minimal	Best for very fast single operations with time between
(b) Compton	Less than (a)	Small (ps and qs only)	Less than (c)	Best for fairly fast continuous processes
(c) Elston (ii)	Minimal	Minimal	Large	Best for slow processes in first-order systems

Fig. 1 Software overheads for the three methods

Notices

NBS parallel computer benchmark collection

The National Bureau of Standards, since its founding, has been concerned with measurement, determining the precise values and metrics for physical phenomena. The NBS has also made significant contributions to metrology in numerous scientific and engineering disciplines. In this tradition, the CMRF (Computer Measurement Research Facility) project at NBS is developing a set of metrics and measurement techniques to characterise the performance of parallel processing systems. As part of that effort, NBS is collecting benchmark programs that represent a variety of applications which are candidates for parallel

processing. NBS solicits benchmark codes from researchers and scientists. Programs which are computationally intensive, I/O intensive, vectorisable or not and from non-numeric as well as from numeric application areas are requested. Especially welcome are programs which have been used to produce timing or speedup data on parallel computers, whose measurement results have been or may be published in the technical literature, and which are in some fairly widely used and higher-level programming language such as FORTRAN, 'C', LISP, Ada, etc. The NBS Collection is intended as a repository of

information about parallel computer benchmarking and will be open to the technical community to borrow from and contribute to as research advances.

Contributions or inquiries should be directed to:

Computer Measurement Research Facility
Institute for Computer Sciences and Technology, Materials Building MS B364,
National Bureau of Standards, Gaithersburg, MD 20899, U.S.A. Telephone:
301-921-3274. (ARPANET: MEASURE @ NBS-VMS).