

AN INTRODUCTION TO HYBRID COMPUTERS AND THEIR APPLICATION TO OPTIMIZATION PROBLEMS

Brian GIRLING

London, England

The author, who is Secretary of the Analog Section of the British Computer Society, describes the way in which some of the logic components of a parallel logic analog computer function, and goes on to discuss the use of such a computer in optimization problems, as for instance in finding the best estimates of the parameters on compartmental analysis of isotopic tracer experiments. The increased power which is gained when an analog computer is coupled to (hybridized with) a general purpose digital computer is then discussed, as shown for example in the increased speed of random-search optimization techniques when an analog computer is used to integrate the differential equations, or the improvement in steepest ascent methods when the parallel logic can be used to keep the hill-climb path continuously on the line of steepest ascent. Finally, the use of a hybrid computer in solving partial differential equations is briefly outlined.

The modern hybrid computer has evolved from the marriage of the general purpose digital computer with its parallel logic analog counterpart. This latter machine is basically an analog computer with some digital logic modules added, giving provision not only for the control of individual analog components but also for the mode of operation of the computer.

The principal advantages of the old analog computer (i.e., one without the parallel logic attachments) had been its speed of operation — it was a true parallel processing machine — and the degree of involvement it required of the problem operator. However, it lacked memory, and the flexibility that only the digital machine could provide. The first advance was to provide the logic, and the second was to link up with a digital computer via an interface. Fig. 1 shows a simplified block diagram of such a linkage, involving the passage of control signals as well as of data or information. This linkage gives a great increase in power, but a parallel logic analog hybrid computer is capable of the solution of significant parameter optimization problems in its own right.

Two hybrid analog elements are of particular interest in this field. The first is the *track-store unit*; the operation of two such units, to produce a parameter sweep signal advancing in discrete steps, is shown in

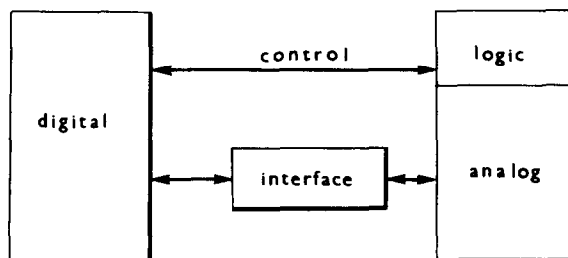


Fig. 1

fig. 2 (the italic letters refer to *logic* signals and the others to *analog* signals). The inputs to the first unit are the analog signals *d* and *f*, the circle on the diagram merely representing a potentiometer used to adjust the magnitude of *d*, and the logic signal *k*. The output of this unit is the analog signal *e*. When the logic signal *k* becomes negative, representing in this case *false*, the unit holds at its output value corresponding to the sum of its inputs at the transition from *true* to *false*, *true* in this case being represented by a zero logic signal. On the other hand, so long as *k* represents *true*, the output is always simply the sum of the inputs with the sign reversed — even though they will, in general, be continuously varying signals.

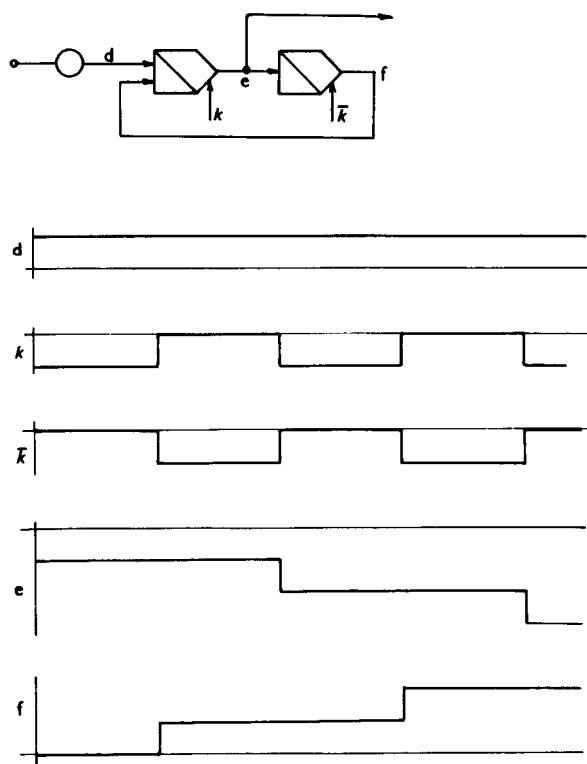


Fig. 2

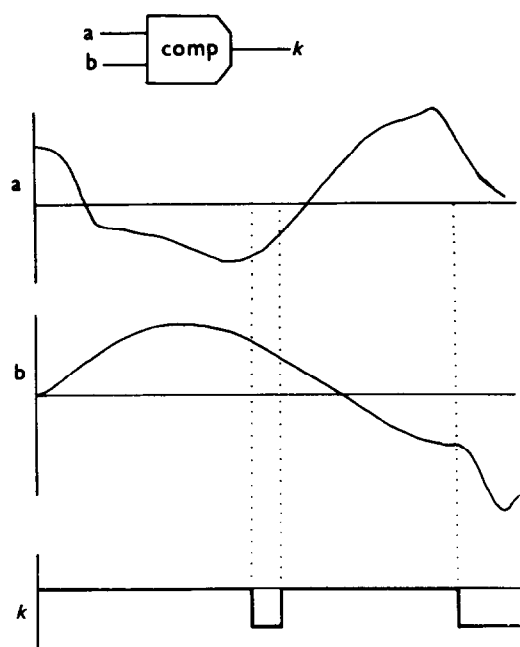


Fig. 3

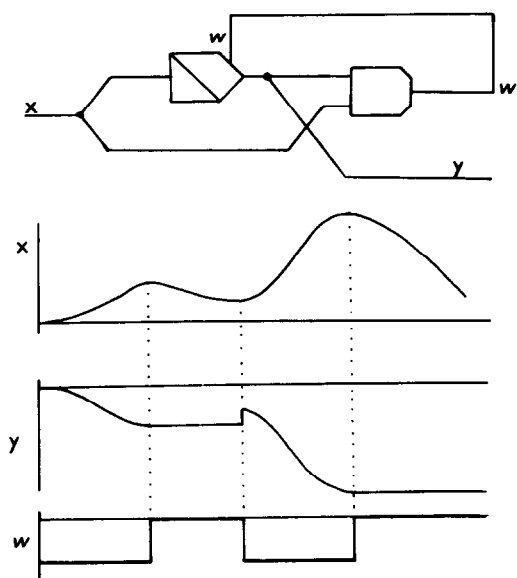


Fig. 4

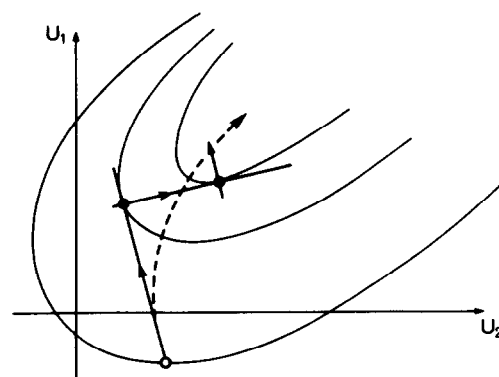


Fig. 5

The sign reversal is the usual one associated with the output from analog components, e.g., if $f = 0.5$ and $d = 0.25$, then $e = -0.75$. The variations of d , e and f with time, as well as that of k and its logical negation \bar{k} , used to control the second track-store unit in opposition to the first, are shown in fig. 2. It can be seen that a constant input d will result in a stepped output at e and f which gives the sweep through the parameter values required.

The second hybrid element of importance in optimization is the *comparator*. This device is shown schematically in fig. 3. It has two analog inputs a and b and one logic output k , which yields the logic value *true* whenever the sum of the inputs is greater than or equal to zero, and *false* otherwise. A track-store unit followed by a comparator is a particularly useful form of maximum-seeking device. When it is connected as in fig. 4, the track-store unit will track the input x as long as x is increasing. This is due to the small propagation delay of the signals as they pass through the track-store unit itself. When x ceases to increase, the comparator changes to *false* and the track-store unit stores the value of the last maximum.

As far as the optimization techniques themselves are concerned, analog optimization methods involving steepest ascents have been employed for some time, and a comparison with their digital counterparts may be useful here. Fig. 5 shows the basic digital approach. Starting from a point on one contour, progress is made along the direction of maximum slope until a maximum occurs in that direction. At that point a decision is made as to the new direction of steepest ascent and the cycle is repeated. Progress towards the maximum takes the form of a series of finite steps, as shown by the arrowed lines of fig. 5. The analog approach, on the other hand, with its continuous adjustment of the variables, produces the path shown by the broken line of the diagram. At every instant the direction must be that of maximum slope unless a constraint intervenes. Various analog devices, such as diode gates, etc., are employed to ensure that when constraints defining a non-feasible region do occur, any attempt to violate them results in the solution curve following the shortest possible path to the nearest boundary (fig. 6a). Further, if a constraint boundary intercepts a steepest ascent path, the solution curve will follow the boundary until it can leave once more by a steepest ascent

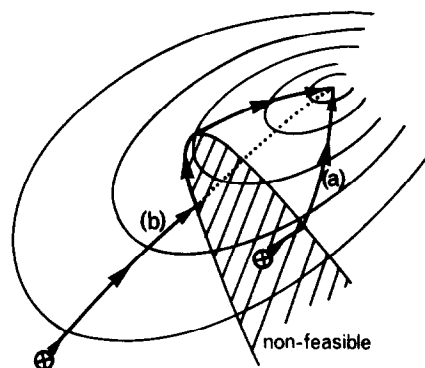


Fig. 6

path, not necessarily the original one (fig. 6b). Fig. 6 shows these last two cases in diagram form.

The straightforward parameter sweep approach can be used together with an iteration technique, but this is of necessity a somewhat time-consuming procedure, and if the hybrid configuration is to compete with the high speed digital computer it must maintain its natural speed differential. Note that *speed* is being used here in the sense of actual computer running time, and does not refer to problem set-up or patching. Repeating a run on an analog or hybrid computer does involve repatching the machine if there has been a substantial lapse of time since the problem was last run. There is no counterpart to the stored digital computer program.

In hybrid optimisation a combination of digital random search coupled with either parameter sweeping or hill-climbing techniques become a feasible method of reducing the time penalty inherent in the purely random search approach. This use of the analog facility to cut down the run time is valid also for other approaches, since a large number of purely digital techniques for optimization revolve round the solution of a system of differential equations or, in certain circumstances, their adjoints. It is in this field that the hybrid computer can come into its own as a means of achieving ultra-high speed integration routines. This is of particular advantage when so-called stiff equations are encountered, as these demand extremely small step lengths when solved digitally, and thus increased solution times which can quite easily exceed the permitted time allocation. The modern hybrid components with their large band-width can handle

these equations quite well, in a fraction of the time required by their digital counterparts.

In conclusion, a rather different problem in which a combination of analog and digital techniques is of great advantage, may be briefly discussed. This is the solution of partial differential equations, such as might occur, in the biochemical field, in diffusion problems. For these equations digital techniques *must* be invoked as the analog computer has only one independent variable, namely time. The method of solution is to replace one or more of the spatial coordinates by finite difference approximation. Fig. 7 illustrates four of the simplest alternatives for the first derivative of the function f , involving a discrete step length h . In three dimensions, two of the space variables U_1 and U_2 will be discretised to give a rectangular mesh, of sides h and k , the third variable U_3 being identified with time. This is shown in fig. 8. Let the problem variables be g and y , where

$$g = g(U_1, U_2, U_3)$$

$$y = y(U_1, U_2, U_3)$$

and the equations are

$$\phi_1 = \phi_1(g, y, U_1, U_2, U_3, \frac{\partial g}{\partial U_1}, \frac{\partial y}{\partial U_1}, \frac{\partial g}{\partial U_2}, \frac{\partial y}{\partial U_2}, \frac{\partial g}{\partial U_3}, \frac{\partial y}{\partial U_3})$$

$$\phi_2 = \phi_2(g, y, U_1, U_2, U_3, \frac{\partial g}{\partial U_1}, \frac{\partial y}{\partial U_1}, \frac{\partial g}{\partial U_2}, \frac{\partial y}{\partial U_2}, \frac{\partial g}{\partial U_3}, \frac{\partial y}{\partial U_3})$$

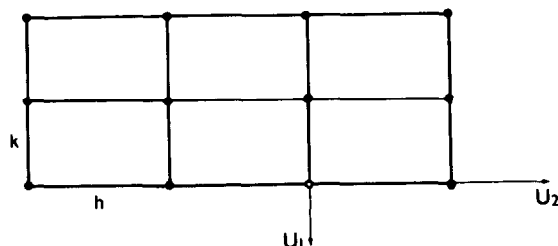


Fig. 8

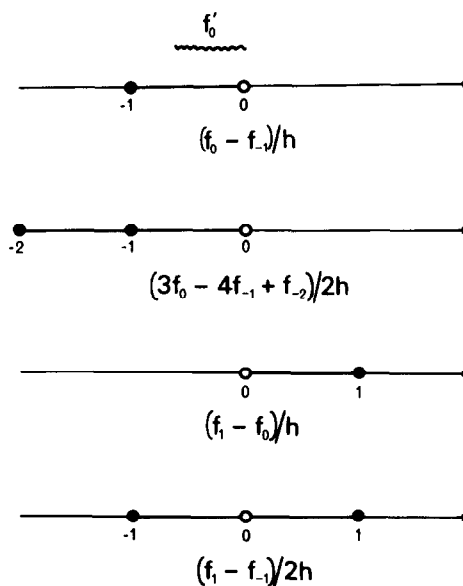


Fig. 7

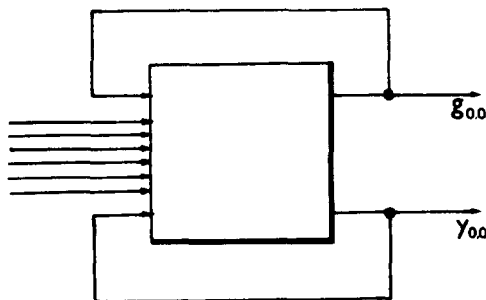


Fig. 9

with suitable initial conditions. Then, replacing all the derivatives with respect to U_1 and U_2 by their finite difference equivalents, a set of functionally identical analog blocks of the type shown in fig. 9 will result, having inputs $g_{r,s}$ and $y_{r,s}$, where $r, s = -1, 0, 1$, and serve to identify the positions adjacent to the solution point. These blocks themselves are connected to form a hardware matrix, and enable the solutions at the mesh points to be determined. Obviously, the more

complex the problem the more hardware is required, but this can now be reduced by time-sharing of analog components under parallel logic and hybrid control.

It is clear that the power of the general-purpose digital computer can be greatly enhanced by the addition of analog elements, under hybrid control, for use as high speed subroutines in the solution of difficult differential equations, thereby improving the turn-round time in an economy-conscious environment.