



# The Differential Scheme for Models of Computation

Howard A. Blair

---

## Abstract

We focus on the metaphor that computation takes place locally, in a space, over time. Three major components of the resulting computational models are (1) the computation space, (2) time, and (3) the admissible local states. Each of these three components may have independently differing topological structure. In particular, each component may be independently chosen to be essentially a discrete space or a continuum. We use a topological idea, neighborhood systems, to rigorously formulate a range of models of computation in accord with the above metaphor, which we call the Differential Scheme. A set equipped with a neighborhood system is a set in which each element  $x$  is associated with a filter of subsets, each member of which is called a neighborhood of  $x$ . Neighborhoods that are neighborhoods of all their members form a topology, but this generally yields a topology too coarse for our purposes. Neighborhood systems in which each filter is principal yield directed graphs, and conversely. Certain neighborhood systems in which each filter is nonprincipal yield continua. An instance  $P$  of the Differential Scheme is specified by choosing sets  $S$ ,  $T$  and  $L$  to serve as space, time, and the set of local states, respectively, and equipping these sets each with a neighborhood system and then specifying a relation in  $(S \times T \times L)$ . The relation is to satisfy certain 'neighborhood constraints' on these components. Continuity is a familiar example of a neighborhood constraint on a function. The relation specifies the flows of  $P$ . It can be seen for example that by holding the neighborhood constraints fixed but varying  $P$  by varying the neighborhood systems, the same neighborhood constraints can on the one hand yield differential equations, and on the other yield Turing machines. One direction of interest with this scheme is to study the variation in flows under changes in neighborhood systems, particularly when flows of more discrete instances of the Differential Scheme are stable within embeddings of these instances in less discrete instances.

---