Geography and Computational Science

Marc P. Armstrong

Department of Geography and Graduate Program in the Applied Mathematical and Computational Sciences, University of Iowa

On many U.S. campuses, a rising tide of scholarly activity, called *computational science*, is being recognized by academic administrators through the establishment of research centers and degree programs (Rice 1994). This nascent discipline has emerged from several traditional fields, including for example, cosmology, geography, and pharmacology, that contain subareas in which computing is the primary tool used to pursue research questions. The common thread that ties together such seemingly disparate activities is a shared focus on the application of advanced computing to problems that heretofore were either intractable, or, in some cases, unimagined. The purpose of this paper is to sketch out the opportunities for geography that lie at the intersection of computational science and geographical modeling.

At the outset, it is important to draw a distinction between computational science and computer science. Though they are related, computational science is concerned with the *application* of computer technology to create knowledge in particular problem domains. Sameh (1995:1), for example, envisions computational science as a multifaceted discipline that can be conceptually represented as a pyramid with an application field (e.g., geography) at the apex. At the four base corners are: (1) algorithms, (2) architectures, (3) system software, and (4) performance evaluation and analysis. O'Leary (1997: 13) articulates a different but related view in which an interdisciplinary and team-oriented computational science rests on the foundational elements of a particular science or engineering discipline, together with mathematics, computer science, and numerical analysis.

Despite some apparent variability in these (and other) views of computational science (cf., Stevenson 1994, 1997), they share a consistent unifying principle: the use of models to gain understanding. While most traditional views of science hold sacred the dyadic link between theory and experimentation, computational scientists have expanded this view to include a separate but equal role for simulation and modeling (Figure 1).

Geographers have been specifying and testing models for decades (Hägerstrand 1967; Chorley and Haggett 1969) and are well positioned to make significant contributions to interdisciplinary computational-science teaching and research initiatives. Despite substantial progress (Longley et al. 1998), however, in many cases, the use of models to support scientific investigations and decisionmaking has been hampered by computational complexity and poor performance. In the next section of this paper, I describe the underlying causes of this computational complexity. Then I focus discussion on the development of synergistic interactions between geography and computational science, placing a particular emphasis on the use of new computer architectures to improve the performance of models and foster their application in an enlarged set of scientific and policy contexts. I next describe how emerging computational technologies will begin to alter approaches to the development of geographical models. Because visualization is an important element of computational science, enabling researchers to gain insights from the results of their numerical computations, in the final section of the paper, I initiate a discussion about how a form of advanced visualization, immersion, is creating a need to rethink aspects of cartography.

The Computational Complexity of Geographic Analyses

Why should geographers care about computational science and high performance computing? The rationale described in this section is not meant to be exhaustive, but rather, to highlight a set of elements that underlie a broader need for concern. It should be noted, however, that while similar arguments, some made more than three decades ago, have been advanced

Annals of the Association of American Geographers, 90(1), 2000, p. 146–156

Published by Blackwell Publishers, 350 Main Street, Malden, MA 02148, and 108 Cowley Road, Oxford, OX4 1JF, UK.

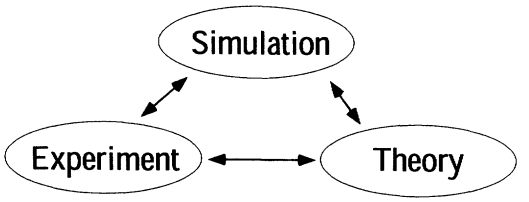

Figure 1. Expanding the relationship between theory and experiment to include simulation and modeling (Source: adapted from Karin and Graham 1998).

(see Gould 1970a; Hägerstrand 1967; Openshaw 1994), the pace of change has recently accelerated and other disciplines with computational-science ties are moving forward rapidly. If geographers fail to make contributions in areas that are fundamentally spatial, other disciplines will (re)develop or (re)invent many concepts and methods that are now known to geographers.

Trends in Information Acquisition

Geographers are witnessing a rapid increase in the volume of information that is either explicitly geographic or that can be converted to geographic information after it is linked to geographic identifiers (e.g., address-matching). Satellite remote sensing instruments, for example, continue to increase in both spatial and radiometric resolution. The original Landsat multispectral scanner had a spatial resolution of approximately 79 m. In contrast, several commercial firms now advertise the availability of data with 1-m spatial resolution. Thus, within the extent of an idealized Landsat MSS pixel, more than 6,000 pixels would exist in an image acquired using one of the new sensor systems. What this means is that an area covered by a mega-pixel Landsat scene ($1024 \times 1024 = 1{,}048{,}576$ pixels) would have 6.29×10^9 pixels in the corresponding area imaged with a 1-m system. These calculations, of course, are for only a single spectral band. With the increasing availability of hyperspectral sensor systems, some with short revisit capabilities, the amount of geographic information being collected can easily increase by orders of magnitude.

Commensurate increases are also being seen in other types of data as in situ sensors record activities such as individual vehicle movements (Summers and Southworth 1998). Disaggregated, individual-level administrative and retail records (e.g., medical information and point-of-sale transactions) are also being linked by addresses to locations using widely available digital street centerline files with associated address ranges (Dueker 1974; Broome and Meixler 1990) and inexpensive desktop software (Beaumont 1989; Openshaw 1993; Armstrong et al. 1999). The simple acts of performing input-output and managing these massive data files requires a considerable amount of processing time. Even more pressing problems occur, however, when these data resources are used in analyses.

Trends in Geographic Modeling

Many types of geographic models are intrinsically computationally intensive. Some involve combinatorial search strategies that explode as a function of problem size. Other types of models have become complex as a consequence of researchers working to capture and incorporate increasingly realistic representations of spatial processes (Ford et al. 1994; Miller 1996). For example, one family of forest-stand simulation models (JABOWA-FORET: Botkin 1993) included a number of biophysical processes in its original formulations but failed to incorporate a component that handled propagule dispersal. When a probabilistic spatial-dispersion module was added, the amount of computation required to execute the model increased considerably (Malanson and Armstrong 1996).

Computer scientists have recognized that spatial-optimization problems, such as the so-called "traveling salesman problem," often require extensive amounts of combinatorial search; they refer to such problems as *NP*-complete (Karp 1976; Garey and Johnson 1979). *NP*-complete problems are distinguished by a sharp increase in run time (greater than polynomial) as a function of problem size. Other spatial-optimization problems are classified *NP*-hard and also require considerable amounts of computation (Kariv and Hakimi 1979). The computational intractability of solution procedures for such problems has been recognized by geographers for decades and is only aggravated by attempts to use disaggregated information (Francis et al. 1999). Gould (1971: 9), in describing the general location-allocation problem, states that it "is, quite literally, brutal" in its computational complexity, and estimates that for a small example problem, approximately one trillion

possibilities would need to be evaluated. More than two decades later, Church et al. (1993:1) make a similar observation when they state that "even some of the most basic models are computationally intractable for all but the smallest problem instances." Hodgson and Berman (1997: 33), for example, report that their billboard-location model, when run for a problem in Edmonton, Alberta, required "several days of CPU time on a small mainframe computer." For these reasons, geographers have been at the forefront of researchers who have developed heuristic methods that first eliminate unlikely possibilities from consideration and then focus only on those parts of a problem that are likely to lead to a solution (e.g., Densham and Rushton 1992; Hodgson 1989; Hodgson and Berman 1997). Heuristic methods, however, are not a panacea. Densham and Armstrong (1994), for example, describe a disaggregate-level optimization problem that would require more than 68 million iterations of a heuristic solution procedure, with each iteration requiring a substantial amount of computation.

Geographers are also now beginning to establish literatures in emerging data-analysis paradigms that are computationally intensive. For example, some researchers have begun to explore data mining and knowledge discovery (Murray and Estivill-Castro 1998). Others have moved away from the use of "classical" inferential statistical methods, since their use with geographic information often violates underlying assumptions (Gould 1970b). Instead, they use bootstrapping methods to assess significance and generate confidence intervals by placing their observed results in the context of an inductively generated reference distribution (Efron 1993; Griffith and Amrhein 1997); computation of a reference distribution for each analysis, however, may require the generation of large numbers (>1,000) of subsamples (see, for example, Openshaw et al. 1988; Gatrell et al. 1996).

Openshaw and Openshaw (1997) and others have written about several other recently developed computational methods, including simulated annealing (Openshaw and Schmidt 1996). Ritter and Hepner (1990), Fisher and Gopal (1993), Hewitson and Crane (1994), and Zhou and Civco (1996) are among researchers who have begun to explore the use of neural networks in geographic applications. Other researchers (Hosage and Goodchild 1986; Armstrong and Bennett 1990; Dibble and Densham 1993) have examined classifier systems and genetic algorithms (Holland et al. 1986; Goldberg 1989; Mitchell 1998) in geographical analysis. Bennett et al. (1999), for example, have remapped the linear "gene" sequences used in traditional genetic algorithms so that they can more directly represent two-dimensional problems. These literatures are increasing in size as additional researchers discover these new analysis paradigms and, as a result, computational requirements are likely to increase. At the present time (and into the foreseeable future), therefore, geographers working in the computational arena will need to concern themselves with the use of high-performance computers.

Parallel Processing and Geographic Models

Moore's Law predicts that the performance of microprocessors will double approximately every eighteen months. This "law" has held true throughout this decade and seems likely to hold in the short term. Despite this remarkable increase in performance, however, computer architects are concerned that their near-future designs are about to reach the physical limits of the materials used in processors (Stone and Cocke 1991; Messina et al. 1998). Matzke (1997), for example, describes problems associated with moving electrical signals rapidly enough over small intraprocessor distances to ensure that all parts can be reached in a single clock-cycle. This latency constraint can be overcome by reducing feature sizes but such reductions are becoming increasingly costly, and difficult to put into production. Moreover, for any given processor design, users have come to expect that increases in clock speed will result in higher performance. Maximum clock-speeds, however, are predicted to increase only by less than a factor of three by 2010 (Messina et al. 1998: 42). Despite this pessimism, there is a considerable amount of university-based and intercorporate cooperative research that is underway with the expressed goal of developing innovative and cost-effective solutions to these and other problems (see, for example, Hamilton, 1999; Patt et al. 1997).

Is there a readily available and less costly way to overcome such performance limitations? The simple answer is: yes. It is clear that raw increases in processor speed fail to account for the

even more dramatic rise in the computational power of high-performance computers observed during the past decade. Messina et al. (1998:37), for example, state that while single-processor performance increased by roughly a factor of 15 during the 1990s, the peak performance of computer *systems* used by computational scientists increased by a factor of 500. These systems typically harness the computational power of collections of computers, operating in parallel, to achieve these high-performance milestones. In fact, of the top twenty high-performance computer systems (last update: June 10, 1999), none had fewer than 64 processors and the system with the highest performance had 9,472 (see http://www.top500.org).

The general principle is that no matter how fast a single processor is, when several are used together (efficiently), an even greater increase in performance can be achieved. Consider, in a metaphorical sense, the experiences that most of us have had in a check-out line at a grocery store. In the distant past, a clerk would search for the price of each item and enter it, using a mechanical adding machine, to keep a running total of purchases. Technical innovations, such as scanners, increased the rate at which price information could be entered, which presumably increased the overall "throughput" of items, and thus customers, at each station. Individual items, however, still require handling and, as a consequence, an effective limit remains on the number of customers that can be reasonably handled by an individual employee. At peak rush times, it is not the typical practice of store management to exhort employees to move more quickly. Instead, clerks open additional check-out lanes, to operate in parallel, thereby spreading the workload (customers) to reduce queue sizes. In the same way, for a total computational workload of a given size, computer systems can finish a job more quickly if tasks are efficiently divided among processors.

Parallelism in Geographic Research

Most geographers are unfamiliar with basic parallel-processing concepts. This is not surprising since there was scant research on applications of parallelism by geographers and planners before 1995. There are several plausible reasons for this. First, though computers with multiple processors have been used widely for decades, they were designed so that typical users were unaware of this fact, and tools were not provided for users to control parallelism. Second, when commercial parallel systems became available, they used control languages that were proprietary to each manufacturer. This trend, which is similar to that followed originally by graphics software (recall CalComp plotting language), meant that parallel software implementations were often limited to use on a single type of computer system. It is only during the past five years that general and portable command languages have been widely adopted. Despite these problems and uncertainties, however, parallelism did attract the attention of some spatial modelers.

In an early paper, Harris (1985) suggested that parallel processing could be fruitfully applied to transportation and land-use models. Sandhu and Marble (1988) explored high-performance computing and the use of parallelism for spatial data processing (Sandhu 1987, 1988). In the following year, Smith et al. (1989) published a paper in which they provided a parallel solution to the weighted-region least-cost path problem, Healey and Desa (1989) evaluated parallel architectures and programming models, and Franklin et al. (1989) described a parallel implementation of a map overlay algorithm. In their discussion, Franklin et al. (1989) address the issue of scalability[1] and report results as they systematically increase the number of processors used in their analysis. Since the early 1990s, however, the pace of publication has increased (see, for example, Mower 1993; Hodgson et al. 1995; Zhang et al. 1998; and the summary in Cramer and Armstrong 1999). Moreover, a special parallel-processing issue of the *International Journal of Geographical Information Systems* has been published (10[6], 1996), and, in 1998, the milestone of a book was achieved (Healey et al. 1998). This increasing level of activity is partially a reflection of broader trends in computing: parallelism has become an important paradigm, access to parallel machines is improving, and languages and architectures are becoming more standardized.

Architectural Futures and Geographical Analyses

There are several ways to achieve parallelism, and hardware vendors have chosen to pur-

sue these different lines of implementation with varying degrees of success. In the early 1990s, the SIMD (single instruction, multiple data) systems at most of the major supercomputing installations employed thousands of simple processors that executed instructions in lock-step (synchronously) on different data. Though they were often effective for programs that processed regular geometrical representations (e.g., remote-sensing images and geographic matrices: [Birkin et al. 1995; Xiong and Marble 1996]), SIMD computers did not perform well in many other problem domains (Armstrong and Marciano 1997). Consequently, SIMD architectures appear to be an evolutionary deadend.

In contrast, MIMD (multiple instruction, multiple data) architectures have persisted in several forms and appear to be long-run survivors. In their basic form, MIMD computers are constructed from a set of largely independent processing elements that access a physical shared memory. Tasks are apportioned to the different processors and results are computed asynchronously (see for example, Armstrong et al. 1994). This model is not without its own problems, but computer architects have worked successfully to overcome limitations. New shared-memory machines, for example, reduce communication bottlenecks with high-speed network switches that permit low-latency communication among processors organized in hierarchically structured clusters (Pfister 1998). In general, researchers who have applied MIMD architectures to geographic problems have met with success: processors are used efficiently, and execution times are reduced substantially.

In 1997, the National Science Foundation phased out funding for its Supercomputer Centers Program and began funding that established its Partnerships for Advanced Computational Infrastructure (PACI) initiative (Smith 1997). Two partnerships, one led by the University of Illinois (National Computational Science Alliance [NCSA]), and the other by the University of California at San Diego (National Partnership for Advance Computational Infrastructure [NPACI]), resulted. Each features computational science as a cornerstone of its activity and places a major emphasis on the development of services that employ distributed shared-memory (DSM) parallel architectures. NCSA, for example, has, as a stated goal, a program to "make shared-memory parallel computing the norm, even at the desktop, over the next five years" (Kennedy et al. 1997: 65). Researchers from both partnerships have contributed to a recently published book in which this vision for the future of computing is more fully developed (Foster and Kesselman 1999). The guiding metaphor is "the grid," specifically to connote the electrical power grid. Computational grids are intended to provide a new infrastructure that supplies dependable, consistent, pervasive, and inexpensive computing.

A key change that arises from grid-based computing is a move away from proprietary, small-volume processors used by obsolete vector supercomputers (e.g., Cray) and massively parallel SIMD machines (e.g., Connection Machine, MasPar) toward off-the-shelf processors. In fact, one DSM approach is referred to as "shared nothing," meaning that each element in the machine is a stand-alone processor with its own memory, disk, and network controller. These distributed elements can be linked into a virtual metasystem that provides a monolithic illusion to the end user (Grimshaw et al. 1997, 1998).

Though some metasystems use ordinary workstations and network connections, others are more formal collections that use special-purpose networks. The NOW (network of workstations) machine at the University of California at Berkeley, for example, is a computational resource of NPACI that consists of approximately one hundred Sun workstations linked with a high-performance network and fast switches to route messages among them (Anderson et al. 1995). Anticipated system upgrades include the addition of a heterogeneous mix of workstations from different manufacturers with some running other types of operating systems (http://now.cs.berkeley.edu/). While early results using this machine in prototype form have been somewhat disappointing (Armstrong and Marciano 1998), the price/performance benefits of using commercial off-the-shelf technology overpower arguments against it, and several initiatives similar to NOW are underway.

One project in particular, called Condor, provides software that implements a high-throughput computing environment constructed from a loosely configured ensemble of workstations (http://www.cs.wisc.edu/condor/). A distinction in this case is made between high-performance computing, which is concerned with reducing response times to as close to interactive as possible, and high-throughput

computing, which is concerned with the delivery of large amounts of processing capacity over long periods of time (e.g., weeks). Condor probes the set of machines that owners have elected to include in its configuration and, when it detects an idle or lightly loaded node, it migrates pending tasks to it, thereby harnessing the otherwise spare cycles (see also Gelernter 1992). If a workstation user once again begins intensive use of an idle machine, the Condor software moves the job elsewhere. For a user to permit their machine to be linked into the Condor configuration, they must have some level of assurance that their own computational requirements will be satisfied and that there will be no adverse outcomes associated with their participation.

Given that several powerful parallel-programming tools are now being developed to support DSM approaches (Saltz et al. 1998), and that the goal of most is to hide low-level details from the user, is it appropriate to wait for fully automated parallel environments? No, because in the short-term, it is unlikely that users will either want, or be able, to remove themselves completely from some level of involvement with the machines on which they compute. Great strides have been made in providing users with tools such as MPI (Snir 1998) and MPI2 (*International Journal of High Performance Computing Applications* 1998) that enable users to write programs that are far more portable than previously possible. And new versions of traditional high-level languages (e.g., High Performance Fortran [HPF]) now handle some of the details of automatic parallelization (Kennedy et al. 1997). Load balancing of geographic information among processors, however, remains an important factor in many parallel applications, since data "locality" can seriously affect performance. The idea is that, if data for a particular processor can be found on a machine that is "near" it in its network arrangement (e.g., in its own memory or that of a machine in a local cluster), then processing will proceed more efficiently (with low latency) than if data must be accessed from a machine that is "far" from it. Grimshaw et al. (1998: 50), for example, state that "it is, after all, at least 30msec from California to Virginia." This invites a restatement of Tobler's oft-invoked law of geography: A program will perform more efficiently if required data can be accessed from a near node. Data, therefore, might be allocated dynamically in uniform blocks, strips, or nonuniform partitionings, such as those used by quadtrees. Research on the application of geographic data structures such as Morton sequences might prove useful in this regard (Ding and Densham 1996; Qian and Peuquet 1997; see also Saltz et al. 1998).

The architectural shift toward distributed parallelism will likely cause researchers to extend their thinking in interesting ways. As decision (and planning) support systems continue to develop, there is a need to develop interactive approaches to the generation and evaluation of alternative solutions to problems (Densham 1994). This need is especially pressing when groups of people are convened to address ill-defined public policy problems (Armstrong, 1994; Churcher and Churcher 1999). In such cases, a group may meet together only for short periods of time but may wish to consider a problem from several perspectives. Often, a computer-produced solution may have some type of identifiable limitation but prove to be a useful point of departure for group discussion. If, however, only one or two computationally complex solutions can be generated during the course of a meeting, group members will be unable to realize substantial benefits from decision-support tools. To circumvent this bottleneck, it is possible to spawn a swarm of software agents (Chorafas 1998; Tecuci 1998; Murch and Johnson 1999) that will autonomously search for spare computer cycles, execute models using different parameters, and report results back to decision-makers. Obviously, research is needed to accomplish this task, and also to help decisionmakers sort through the complexity of evaluating competing alternatives.

Researchers will also be required to rethink and, in some cases recast, their approaches to computer-based modeling and analysis if they wish to reap the performance benefits of new parallel environments. Modelers have made a substantial investment in formulating computer code so that it can run efficiently on the dominant, sequential architecture of the past several decades. It is somewhat ironic that these highly optimized algorithms will not perform well in parallel environments if they rely on intricate data structures that preclude independence of program elements—this is anathema to parallel efficiency. Armstrong and Densham (1992), for example, suggest that one simple, but relatively slow, locational modeling heuris-

tic would be much easier to implement and execute in parallel environments than other widely used but more complex approaches. Issues such as this need additional research as geographers and computational scientists turn their attention to spatially explicit models that execute on state-of-the-art, distributed parallel architectures.

Visualization, Computational Science, and Cartography

Researchers working with complex models often wish to visualize their work to "see" if results make intuitive and theoretical sense. Many model applications in computational science have a geographic component and thus, maps are often needed. Advances in visualization have taken place on several fronts, but few are more immediately and simultaneously visceral and informative than immersion. New technologies enable researchers to virtually walk around inside representations of their models to gain a better understanding of their data and modeling results, including the interrelationships among variables and the various parameterizations used (Foster and Kesselman 1999). Though such approaches to visualization are still somewhat exotic, many universities have acquired projection-based virtual-reality environments that can be used to visualize spatial processes (Reed et al. 1997) and support virtual exploratory analyses.

It is interesting to note, however, that while cartographers have worked diligently during the past several decades to understand map communication and use, they have largely operated under an "abstractive" paradigm: maps are abstractions of reality that are created through a process of applying a set of limiting filters. This paradigm has driven a considerable amount of research in areas such as symbolization and generalization. This work is valuable and must be continued, but there is a tension emerging between the abstractions of cartographers and the virtual, augmented realities that are now being created in advanced visualization laboratories. Research needs to be conducted that will reconcile these views. It is likely, for example, that hierarchical generalization research and generalization algorithms (McMaster and Shea 1992) will prove to be useful to researchers interested in developing distributed immersive collaboratories (Kouzes et al. 1996), and yet, visualization researchers seem to be unaware of this work (Song and Norman 1994).

Conclusion

As we begin the year 7D0 (hexadecimal), current trends suggest that the computational requirements of geographic research will continue to increase. New sources of disaggregated and high-resolution data are becoming widely available; we are now awash in data resources, and this tide will not ebb. Geographers and computational scientists will wish to use these data in increasingly detailed dynamic spatial models. In addition, new computationally intensive computing paradigms are making inroads in the many areas of geography that routinely use computing to help researchers investigate research problems. Geographers, for example, are beginning to adopt computational approaches such as genetic algorithms (and other types of evolutionary programming), simulated annealing, neural networks, and data mining. Most of these new paradigms hold something in common: they are explicitly parallel.

This is fortuitous since high-performance computing is now synonymous with parallel processing. Parallel architectures are becoming increasing decentralized as off-the-shelf workstations are being linked into flexible clusters onto which parts of large problems can be allocated to independent processing nodes. This provides a cost-effective approach to parallelism that is within the reach of researchers at most universities. This innovation, when coupled with new machine-independent approaches to parallel programming, and broad access to high-speed networks (http://www.internet2.edu) and visualization tools, will provide geographers with an exciting testbed for the design and implementation of parallel, high resolution, spatially explicit simulation models and new classes of distributed computationally intensive geographic models with interacting components. As geographers increasingly come to employ parallel architectures in their work, they will also, perhaps, forge alliances with the growing number of computational scientists who are developing and evaluating spatial models. This interaction will be beneficial to the interdisciplinary computational science community, will strengthen the position of geographers in a variety of academic and research settings, and will

lead to an improved understanding of complex spatial processes.

Acknowledgments

I would like to thank Harvey Miller, Gerard Rushton, and Claire Ellen Pavlik for their comments on earlier drafts of this paper. Partial support was provided by the National Science Foundation (DUE 97-50537) and the National Aeronautics and Space Administration (NAG5-6670).

Note

1. Scalability is an important issue in parallel implementations, for if a program is applied to a dataset and does not produce results in less time as additional processors are introduced, computational resources are not used efficiently. A measure of speedup, defined as the run time required to execute a program with one processor, divided by the run time required with *n* processors, is often used to help assess the scalability of an implementation. Speedups are usually computed across the range of available processors (e.g., 1, 2, 4, 8, and 16 processors in a 16-processor environment) and graphed with the expectation that for a "perfectly efficient" program, the slope of the graph will be one if unit spacing is used on both axes (see, for example, Armstrong and Marciano 1994). When systematic departures from this expectation are observed, changes are often made in an attempt to improve software performance.

References

Anderson, T.; Culler, D.; and Patterson, D. 1995. A Case for NOW (Networks of Workstations). *IEEE Micro* 15: 54–64.

Armstrong, M. 1994. Requirements for the Development of GIS-Based Group Decision Support Systems. *Journal of the American Society for Information Science* 45: 669–77.

——— and Bennett, D. 1990. A Bit-Mapped Classifier for Groundwater Quality Assessment. *Computers and Geosciences* 16: 811–32.

——— and Densham, P. 1992. Domain Decomposition for Parallel Processing of Spatial Problems. *Computers, Environment and Urban Systems* 16: 497–513.

——— and Marciano, R. 1994. Inverse Distance Weighted Spatial Interpolation Using a Parallel Supercomputer. *Photogrammetric Engineering and Remote Sensing* 60: 1097–1103.

——— and ———. 1997. Massively Parallel Strategies for Local Spatial Interpolation. *Computers & Geosciences* 23: 859–67.

——— and ———. 1998. A Network of Workstations (NOW) Approach to Spatial Data Analysis: The Case of Distributed Parallel Interpolation. *Proceedings of the Eighth International Symposium on Spatial Data Handling*, pp. 287–96. Burnaby, BC: International Geographical Union.

Armstrong, M.; Pavlik, C.; and Marciano, R. 1994. Parallel Processing of Spatial Statistics. *Computers & Geosciences* 20: 91–104.

Armstrong, M.; Rushton, G.; and Zimmerman, D. 1999. Geographically Masking Health Data to Preserve Confidentiality. *Statistics in Medicine* 18: 497–525.

Beaumont, J. 1989. Geodemographics in Practice: Developments in Britain and Europe. *Environment and Planning A* 21: 587–604.

Bennett, D.; Wade, G.; and Armstrong, M. 1999. Exploring the Solution Space of Semistructured Geographical Problems Using Genetic Algorithms. *Transactions in GIS* 3: 51–71.

Birkin, M.; Clarke, M.; and George, F. 1995. The Use of Parallel Computers to Solve Nonlinear Optimisation Problems: An Application to Network Planning. *Environment and Planning A* 27: 1049–68.

Botkin, D. 1993. *Forest Dynamics*. Oxford: Oxford University Press.

Broome, F., and Meixler, D. 1990. The TIGER Data Base Structure. *Cartography and Geographic Information Systems* 17: 39–47.

Chorafas, D. 1998. *Agent Technology Handbook*. New York: McGraw-Hill.

Chorley, R., and Haggett, P. 1969. *Models in Geography*. London: Methuen.

Church, R.; Current, J.; and Eiselt, M. 1993. Editorial. *Location Science* 1:1.

Churcher, C., and Churcher, N. 1999. Realtime Conferencing in GIS. *Transactions in GIS* 3: 23–30.

Cramer, B., and Armstrong, M. 1999. An Evaluation of Domain Decomposition Strategies for Parallel Spatial Interpolation of Surfaces. *Geographical Analysis* 31: 148–68.

Densham, P. 1994. Integrating GIS and Spatial Modelling: Visual Interactive Modelling and Location Selection. *Geographical Systems* 1: 203–19.

——— and Armstrong, M. 1994. A Heterogeneous Processing Approach to Spatial Decision Support Systems. In *Advances in GIS Research*, ed. T. C. Waugh and R. G. Healey, vol. 1, pp. 29–45. London: Taylor and Francis Publishers.

——— and Rushton, G. 1992. Strategies for Solving Large Location-Allocation Problems by Heuristic Methods. *Environment and Planning A* 24: 289–304.

Dibble, C., and Densham, P. 1993. Generating Inter-

esting Alternatives in GIS and SDSS Using Genetic Algorithms. *Proceedings of GIS/LIS'93*. Bethesda, MD: American Congress on Surveying and Mapping.

Ding, Y., and Densham, P. 1996. Spatial Strategies for Parallel Spatial Modelling. *International Journal of Geographical Information Systems* 10: 669–98.

Dueker, K. 1974. Urban Geocoding. *Annals of the Association of American Geographers* 64: 318–25.

Efron, B. 1993. *An Introduction to the Bootstrap*. New York: Chapman & Hall.

Fischer, M., and Gopal, S. 1993. Neurocomputing—A New Paradigm for Geographic Information Processing. *Environment and Planning A* 25: 757–60.

Ford, R.; Running, S.; and Nemani, R. 1994. A Modular System for Scalable Ecological Modeling. *Computational Science and Engineering* 1: 32–44.

Foster, I., and Kesselman, C. 1999. *The Grid: Blueprint for a New Computing Infrastructure*. San Francisco, CA: Morgan Kaufman.

Francis, R.; Lowe, T.; Rushton, G.; and Rayco, M. 1999. A Synthesis of Aggregation Methods for Multi-Facility Location Problems: Strategies for Containing Error. *Geographical Analysis* 31: 67–87.

Franklin, R. 1989. Uniform Grids: A Technique for Intersection Detection on Serial and Parallel Machines. *Proceedings of Auto-Carto 9*, pp. 100–109. Bethesda, MD: American Congress on Surveying and Mapping.

Garey, M., and Johnson, D. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: W. H. Freeman and Co.

Gatrell, A.; Bailey, T.; Diggle, P.; and Rowlingson, B. 1996. Spatial Point Pattern Analysis and Its Application in Geographical Epidemiology. *Transactions of the Institute of British Geographers* 21: 256–74.

Gelernter, D. 1992. *Mirror Worlds: Or the Day Software Puts the Universe in a Shoebox . . . How It Will Happen and What It Will Mean*. New York: Oxford University Press.

Goldberg, D. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley.

Gould, P. 1970a. Computers and Spatial Analysis: Extensions of Geographic Research. *Geoforum* 1: 53–69.

———. 1970b. Is *Statistix Inferens* the Geographical Name for a Wild Goose? *Economic Geography (supplement)* 46: 439–48.

———. 1971. Introduction. In *Multiple Location Analysis*, G. Tornqvist, S. Nordbeck, B. Rystedt, and P. Gould, *Lund Studies in Geography, Series C. General, Mathematical and Regional Geography* 12. Lund: Royal University of Sweden.

Griffith, D., and Amrhein, C. 1997. *Multivariate Statistical Analysis for Geographers*. Upper Saddle River, NJ: Prentice Hall.

Grimshaw, A.; Wulf, W.; and the Legion Team. 1997. The Legion Vision of a Worldwide Virtual Computer. *Communications of the Association for Computing Machinery* 40: 39–45.

Grimshaw, A.; Ferrari, A.; Lindahl, G.; and Holcomb, K. 1998. Metasystems. *Communications of the Association for Computing Machinery* 41: 46–55.

Hägerstrand, T. 1967. The Computer and the Geographer. *Transactions of the Institute of British Geographers* 42: 1–19.

Hamilton, S. 1999. Taking Moore's Law into the Next Century. *Computer* 32: 43–48.

Harris, B. 1985. Some Notes on Parallel Computing: With Special Reference to Transportation and Land Use Modelling. *Environment and Planning A* 17: 1275–78.

Healey, R., and Desa, G. 1989. Transputer Based Parallel Processing for GIS Analysis: Problems and Potentialities. *Proceedings of the 9th International Symposium on Computer Assisted Cartography*, pp. 90–99. Bethesda, MD: American Congress on Surveying and Mapping.

Healey, R.; Dowers, S.; Gittings, B.; and Mineter, M. 1998. *Parallel Processing Algorithms for GIS*. Bristol, PA: Taylor and Francis.

Hewitson, B., and Crane, R. 1994. *Neural Nets: Applications in Geography*. Boston: Kluwer Academic Publishers.

Hodgson, M.E. 1989. Searching Methods for Rapid Grid Interpolation. *The Professional Geographer* 41: 51–61.

———; Cheng, Y.; Coleman, P.; and Durfee, R., 1995. Computational GIS Burdens. *Geo Info Systems* 5: 28–37.

Hodgson, M.J., and Berman, O. 1997. A Billboard Location Model. *Geographical and Environmental Modelling* 1: 25–45.

Holland, J.; Holyoak, K.; Nisbett, R.; and Thagard, P. 1986. *Induction: Processes of Inference, Learning and Discovery*. Cambridge, MA: MIT Press.

Hosage, C., and Goodchild, M. 1986. Discrete Space Location-Allocation Solutions from Genetic Algorithms. *Annals of Operations Research* 6: 35–46.

Karin, S., and Graham, S. 1998. The High Performance Computing Continuum. *Communications of the Association for Computing Machinery* 41: 32–35.

Kariv, O., and Hakimi, S. 1979. An Algorithmic Approach to Network Optimization Problems II: The p-Medians. *SIAM Journal of Applied Mathematics* 37: 539–60.

Karp, R. 1976. The Probabilistic Analysis of Some Combinatorial Search Algorithms. In *Algorithms and Complexity*, ed. J. F. Traub, pp. 1–20. New York: Academic Press.

Kennedy, K.; Bender, C.; Connolly, J.; Hennessy, J.;

Vernon, M.; and Smarr, L. 1997. A Nationwide Parallel Computing Environment. *Communications of the Association for Computing Machinery* 40: 63–72.

Kouzes, R.; Myers, J.; and Wulf, W. 1996. Collaboratories: Doing Science on the Internet. *Computer* 29: 40–46.

Longley, P.; Brooks, S.; McDonnell, R.; and MacMillan, B. 1998. *Geocomputation: A Primer*. New York: John Wiley & Sons.

Malanson, G., and Armstrong, M. 1996. Dispersal Probability and Forest Diversity in a Fragmented Landscape. *Ecological Modelling* 87: 91–102.

Matzke, D. 1997. Will Physical Scalability Sabotage Performance Gain? *Computer* 30: 37–39.

McMaster, R., and Shea, K. 1992. *Generalization in Digital Cartography*. Washington: Association of American Geographers.

Messina, P.; Culler, D.; Pfeiffer, W.; Martin, W.; Oden, J.; and Smith, G. 1998. Architecture. *Communications of the Association for Computing Machinery* 41: 36–44.

Miller, H. 1996. GIS and Geometric Representation in Facility Location Problems. *International Journal of Geographical Information Systems* 10: 791–816.

Mitchell, M. 1998. *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press.

Mower, J. 1993. Automated Feature and Name Placement on Parallel Computers. *Cartography and Geographic Information Systems* 20: 69–82.

MPI2: A Message-Passing Interface Standard. 1998. *International Journal of High Performance Computing Applications* 12 (1–2).

Murch, R., and Johnson, T. 1999. *Intelligent Software Agents*. Upper Saddle River, NJ: Prentice Hall.

Murray, A., and Estivill-Castro, V. 1998. Cluster Discovery Techniques for Exploratory Spatial Analysis. *International Journal of Geographical Information Science* 12: 431–43.

O'Leary, D. 1997. Teamwork: Computational Science and Applied Mathematics. *Computational Science and Engineering* 4: 13–18.

Openshaw, S. 1993. Over Twenty Years of Data Handling and Computing in Environment and Planning. *Environment and Planning A* (anniversary issue): 69–78.

———. 1994. Computational Human Geography: Towards a Research Agenda. *Environment and Planning A* 26: 499–505.

———, and Openshaw, C. 1997. *Artificial Intelligence in Geography*. New York: John Wiley.

——— and Schmidt, J. 1996. Parallel Simulated Annealing and Genetic Algorithms for Reengineering Zoning Systems. *Geographical Systems* 3: 201–20.

———; Charlton, M.; and Craft, A. 1988. Searching for Leukaemia Clusters Using a Geographical Analysis Machine. *Papers and Proceedings of the Regional Science Association* 64: 95–106.

Patt, Y.; Patel, S.; Evers, M.; Friendly, D.; and Stark, J. 1997. One Billion Transistor, One Uniprocessor, One Chip. *Computer* 30: 51–57.

Pfister, G. 1998. *In Search of Clusters*, 2nd ed. Upper Saddle River, NJ: Prentice Hall.

Qian, L., and Peuquet, D. 1997. A Systematic Strategy for High Performance GIS. *Proceedings of the Thirteenth Symposium on Automated Cartography*, Auto-Carto 13, pp. 145–54. Bethesda, MD: American Congress on Surveying and Mapping.

Reed, D.; Giles, R.; and Catlett, C. 1997. Distributed Data and Immersive Collaboration. *Communications of the Association for Computing Machinery* 40: 39–48.

Rice, J. 1994. Academic Programs in Computational Science and Engineering. *Computational Science and Engineering* 1: 13–21.

Ritter, N., and Hepner, G. 1990. Application of an Artificial Neural Network to Land-Cover Classification of Thematic Mapper Imagery. *Computers & Geosciences* 16: 873–80.

Saltz, J.; Sussman, A.; Graham, S.; Demmel, J.; Baden, S.; and Dongarra, J. 1998. Programming Tools and Environments. *Communications of the Association for Computing Machinery* 41: 64–73.

Sameh, A. 1995. An Auspicious Beginning. *Computational Science and Engineering* 2: 1, 96.

Sandhu, J. 1988. An Overview of Supercomputing, Parallel Processing and Its Implications for Spatial Algorithms. Masters Thesis, Department of Geography, State University of New York at Buffalo, Amherst, NY.

———. 1987. New Computer Architectures and Their Applications in Geography. *Proceedings of the First Latin American Conference on Computers in Geography*, pp. 601–26. San Jose, Costa Rica.

——— and Marble, D. 1988. An Investigation into the Utility of the Cray X-MP Supercomputer for Handling Spatial Data. *Proceedings of the Third International Symposium on Spatial Data Handling*, pp. 253–67. Columbus, OH: International Geographical Union Commission on Geographical Data Sensing and Processing.

Smith, P. 1997. The NSF Partnerships and the Tradition of U.S. Science and Engineering. *Communications of the Association for Computing Machinery* 40: 35–37.

Smith, T.; Peng, G.; and Gahinet, P. 1989. Asynchronous, Iterative, and Parallel Procedures for Solving the Weighted-Region Least Cost Path Problem. *Geographical Analysis* 21: 147–66.

Snir, M. 1998. *MPI—The Complete Reference*. Cambridge, MA: MIT Press.

Song, D., and Norman, M. 1994. Looking In, Looking Out: Exploring Multiscale Data with Virtual Reality. *Computational Science and Engineering* 1: 53–64.

Stevenson, D. 1994. Science, Computational Science, and Computer Science: At a Crossroads.

Communications of the Association for Computing Machinery 37: 85–96.

———. 1997. How Goes CSE? Thoughts on the IEEE CS Workshop at Purdue. *Computational Science and Engineering* 4: 49–54.

Stone, H., and Cocke, J. 1991. Computer Architecture in the 1990s. *Computer* 24: 30–38.

Summers, M., and Southworth, F. 1998. Design of a Testbed to Assess Alternative Travel Behavior Models within an Intelligent Transportation System Architecture. *Geographical Systems* 5: 91–115.

Tecuci, G. 1998. *Building Intelligent Agents*. San Diego, CA: Academic Press.

Xiong, D., and Marble, D. 1996. Strategies for Real-Time Spatial Analysis Using Massively Parallel SIMD Computers: An Application to Urban Traffic Flow Analysis. *International Journal of Geographical Information Systems* 10: 769–89.

Zhang, Z.; Kalluri, S.; JaJa, J.; Ling, S.; and Townshend, J. 1998. Models and High-Performance Algorithms for Global BRDF Retrieval. *Computational Science and Engineering* 5: 16–29.

Zhou, J., and Civco, D. 1996. Using Genetic Learning Neural Networks for Spatial Decision Making in GIS. *Photogrammetric Engineering & Remote Sensing* 62: 1287–95.

Correspondence: Department of Geography, University of Iowa, Iowa City, IA 52242, email marc-armstrong@uiowa.edu.