

Distributed Parallel Processing for Multidimensional Maximum Entropy Reconstruction

Kuo-Bin Li, Alan S. Stern, and Jeffrey C. Hoch¹

Rowland Institute for Science, 100 Edwin H. Land Boulevard, Cambridge, Massachusetts 02142

Received March 20, 1998; revised May 11, 1998

We have developed a two-dimensional maximum entropy spectrum reconstruction program designed to run in parallel on workstation clusters. Test reconstructions of planes extracted from a three-dimensional NMR data set indicate that the parallel speedup is nearly equal to the number of processors provided that the individual processors have comparable performance and that there are at least as many planes as processors. The program also works well in a typical laboratory setting consisting of heterogeneous workstations. © 1998 Academic Press

Key Words: maximum entropy reconstruction; parallel processing; spectrum analysis.

Maximum entropy reconstruction (MaxEnt) is a method of spectrum analysis that avoids some of the shortcomings of the discrete Fourier transform (1–3). Other methods with similar abilities to obtain high resolution spectral estimates from short data records include maximum likelihood reconstruction (4, 5) and methods based on linear prediction (6, 7). MaxEnt is more versatile, however, because it is applicable to spectra containing arbitrary lineshapes and it does not require data to be sampled at uniform intervals (8, 9). The ability to reconstruct spectra from data collected at arbitrary times affords a number of advantages. For example, nonlinear sampling permits spectra to be obtained using less measuring time than conventional linear sampling without loss of sensitivity or resolution, or it can be used to obtain higher resolution or sensitivity in the same measuring time (8–11). These benefits accrue mainly in the indirect dimensions of multidimensional experiments, where an increase in the number of sampled times leads to a proportionate increase in the total measuring time.

Nonlinear sampling applied simultaneously to more than one indirect dimension necessitates computation of a multidimensional MaxEnt reconstruction. Because of the significant computational demands, our previous applications of this approach utilized special-purpose parallel computers (10, 11). Not many NMR laboratories have routine access to comparable resources; indeed, the computing environment in most NMR laboratories is more likely to consist of workstations and personal computers from different vendors and spanning several generations. In this com-

munication we report the development of distributed parallel software for computing multidimensional MaxEnt reconstructions on both heterogeneous and homogeneous networks of workstations and PCs. Our program is a straightforward implementation using standard techniques of coarse-grained parallelism. It makes the benefits of multidimensional MaxEnt reconstruction and nonlinear sampling accessible to a broader range of NMR laboratories. The software is available without charge from the authors (hoch@rowland.org).

Our program is based on the “Cambridge” algorithm (12); extension to multiple dimensions is straightforward (13). Typically, six to eight discrete Fourier transformations as well as several large inner products are required per iteration, and convergence occurs within 40 iterations. Intermediate storage on the order of 16 times the size of the spectrum is required for efficient computation. For even a modest-resolution spectrum consisting of 2048×2048 (hypercomplex) points, the intermediate storage requirements for two-dimensional MaxEnt reconstruction become enormous: 1 gigabyte. Efficient two-dimensional reconstruction of such spectra remains the province of supercomputer-class machines. For individual planes of three- or higher-dimensional spectra, however, the storage requirements are more modest. Two-dimensional reconstruction of a 256×256 hypercomplex plane requires up to 16 megabytes of intermediate storage, a size that comfortably fits the amount of random-access memory available on typical workstations and personal computers. Furthermore, using the “constant- λ ” algorithm (14), it is possible to compute a series of single-plane reconstructions that is equivalent to the more general 3D reconstruction.

While the reconstruction of an individual plane using a workstation is practical, the task of reconstructing all of the planes comprising a multidimensional spectrum remains daunting. When more than one computer is available, a logical approach is to distribute the planes among the available computers. Significant time savings using this distributed parallel scheme can be achieved if the time necessary for computing the reconstruction of an individual plane is much greater than the time necessary for transmitting the input data and the final results between computers. For a 256×256 hypercomplex plane, the amount of data that must be transmitted is less than 2 megabytes: The size of the input data is usually less than that of the reconstructed spectrum, and the

¹ To whom correspondence should be addressed. Fax: 617-497-4627. E-mail: hoch@rowland.org.

TABLE 1
Processor Architectures and Configurations of the Computers in the Heterogeneous Cluster^a

Processor	Configurations	Output data size		
		64 × 64	128 × 128	256 × 256
195 MHz R10000	1 MB L2 cache, 256 MB main memory, IRIX 6.2	3.4 s	10 s	120 s
175 MHz R10000	1 MB L2, 256 MB main memory, IRIX 6.3	3.9	14	130
180 MHz R5000	512 KB L2, 128 MB main memory, IRIX 6.3	8.4	31	140
166 MHz Pentium	256 KB L2, 32 MB main memory, Windows 95	16	36	140
90 MHz Pentium	256 KB L2, 128 MB main memory, Windows 95	29	69	290
25 MHz RS6000	96 MB main memory, AIX 3.2	39	120	610
33 MHz R3000	80 MB main memory, IRIX 4.0.5	60	170	660
20 MHz SPARC 1	24 MB main memory, SunOS 4.1.3	160	450	1700

^a Elapsed times are shown for three single-plane reconstructions.

intermediate results do not need to be transmitted. For a conservative transfer rate of 0.1 megabytes/s, the time required for communication is less than 20 s. Computation of the two-dimensional MaxEnt reconstruction requires several minutes on a modern workstation; MaxEnt reconstruction of individual planes of a multidimensional spectrum is thus a reasonable candidate for distributed parallel processing.

Our implementation of a distributed parallel algorithm for two-dimensional MaxEnt reconstruction uses a master–slave protocol. One computer is designated the master and is responsible for reading in the data, distributing it among the available slave computers, collecting the results, and writing them out. The master is responsible for load balancing, so that a computer that is slow to complete its task, either by reason of time-sharing with other users or because the computer is intrinsically slower, will not adversely affect the time to complete the overall reconstruction. Tasks (i.e., planes) are assigned to CPUs as they become idle, and after all the tasks have been assigned, CPUs that become idle are assigned tasks that are underway but not yet complete. Thus, a faster processor may complete a task that was initially assigned to a slower CPU. We utilized the Parallel Virtual Machine (PVM) library of routines (15) for managing communication between the master and slaves; this software is freely available and runs on a wide variety of operating systems and computer architectures.

Our program was tested on two different workstation configurations. The homogeneous cluster consisted of up to eight Silicon Graphics Power Indigo2 workstations, each containing one MIPS R10000 CPU clocked at 195 MHz, and interconnected via Asynchronous Transfer Mode (ATM) over 155 Mb/s optical fiber using a Fore Systems ASX-200 switch. Each individual processor had 1 MB of L2 cache and 128 MB of main memory and ran under IRIX 6.2. The measured communication bandwidth (using PVM library calls) was about 8.0 MB/s. The heterogeneous cluster consisted of up to eight computers (six different processor architectures) interconnected via standard 10 Mb/s Ethernet. The measured bandwidth was about 1.0 MB/s. The CPU architecture and operating system of each computer are listed in Table 1, together with elapsed times for reconstructing a single plane.

Our test data consisted of 32 planes extracted from a three-dimensional NMR data set. In the first test, the input data size for each plane was 50 × 64 and the output data size was 64 × 64 hypercomplex points. In the second and the third tests, the input data size was 50 × 78 while the output data sizes were 128 × 128 and 256 × 256, respectively. The amount of intermediate storage required for the three tests, as measured under IRIX 6.2, was 2, 4, and 10 MB, respectively. Each test was run eight times, using different numbers of processors. For the heterogeneous cluster, processors were added in the order shown in Table 1, which is roughly the order of CPU speed. The elapsed times and the speedups (defined as the ratio of the elapsed time for a single processor to the elapsed time for N processors) are shown in Fig. 1. For the homogeneous cluster, the speedup is nearly equal to the number of CPUs, which means the efficiency of CPU utilization is close to 100%. In contrast, the speedup for the heterogeneous cluster appears to reach a plateau. Comparison with Table 1 shows that the plateau occurs when the added processors are significantly slower than those already in use. As a result of the load balancing, faster CPUs complete their own tasks and then are available to complete the tasks initially assigned to the slower CPUs. In effect, the slowest CPUs hardly contribute to the calculation at all. It should also be noted that there will be no additional speedup when the number of processors exceeds the number of planes.

In these tests the elapsed time for the homogeneous cluster is always shorter than for the heterogeneous cluster, which is to be expected since each of the processors in the heterogeneous cluster, except for the first one, is slower than the processors in the homogeneous cluster. For both clusters, the communication overhead never exceeded 5% of the total time and diminished as the problem size increased. This overhead is sufficiently small that there is little reason to consider other schemes for partitioning the computations. Indeed, a fine-grained parallelism in which each plane is handled by several processors would be expected to increase the communication overhead (the nonlocality of the Fourier transform means that the entire interferogram would have to be transmitted multiple times among all the processors handling that plane). Finally, we note that at the time we ran the tests, the workstations were lightly

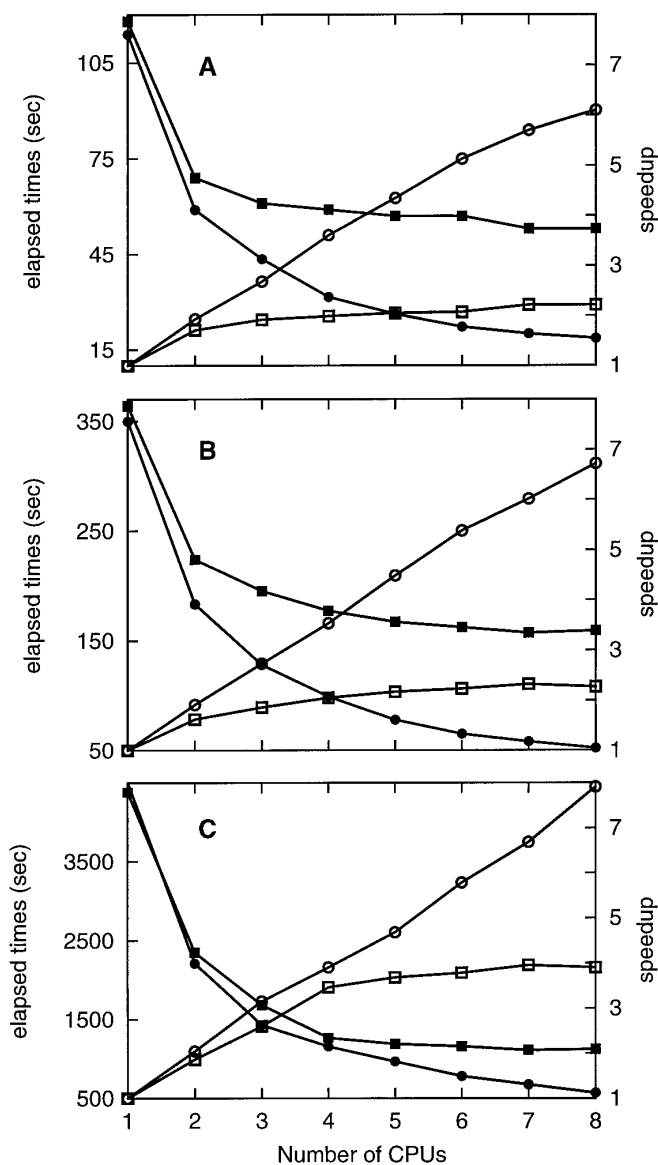


FIG. 1. Elapsed times and speedups for the MaxEnt reconstructions of 32 planes as a function of the number of processors. Elapsed times for the heterogeneous cluster (■), for the homogeneous cluster (●). Speedups for the heterogeneous cluster (□), for the homogeneous cluster (○). Output data size is (A) 64×64 hypercomplex points, (B) 128×128 , (C) 256×256 .

loaded. Nevertheless, they are all multitasking systems and the networks connecting them are also shared, both of which limit the precision of our test results.

We have demonstrated the feasibility of performing maximum entropy reconstructions in parallel on clusters of workstations and PCs. CPU utilization is nearly optimal, with slower CPUs contributing relatively less to the overall performance, as one would expect. Nevertheless, using this approach, the eclectic collection of computers found in a typical labora-

tory can be used to perform difficult multidimensional maximum entropy reconstructions.

ACKNOWLEDGMENTS

This work was supported by grants from NSF (MCB-9527181) and NIH (GM-47467), and by The Rowland Institute for Science.

REFERENCES

1. S. Sibisi, J. Skilling, R. G. Brereton, E. D. Laue, and J. Staunton, Maximum entropy signal processing in practical NMR spectroscopy, *Nature* **311**, 446–447 (1984).
2. J. C. Hoch and A. S. Stern, in "Encyclopedia of Nuclear Magnetic Resonance" (D. M. Grant and R. K. Harris, Eds.), Vol. 5, pp. 2980–2988, John Wiley & Sons, Chichester, UK (1996).
3. M.-A. Delsuc, in "Maximum Entropy and Bayesian Methods" (J. Skilling, Ed.), p. 285, Kluwer Academic, Dordrecht, The Netherlands (1989).
4. G. L. Bretthorst, Bayesian analysis. I. Parameter estimation using quadrature NMR models, *J. Magn. Reson.* **88**, 533–551 (1990).
5. R. A. Chylla and J. L. Markley, Improved frequency resolution in multidimensional constant-time experiments by multidimensional Bayesian analysis, *J. Biomol. NMR* **3**, 515–533 (1993).
6. H. Barkhuijsen, R. de Beer, W. M. M. J. Bovee, and D. van Ormondt, Retrieval of frequencies, amplitudes, damping factors, and phases from time-domain signals using a linear least-square procedure, *J. Magn. Reson.* **61**, 465–481 (1985).
7. H. Barkhuijsen, R. de Beer, and D. van Ormondt, Improved algorithm for noniterative time-domain model fitting to exponentially damped magnetic resonance signals, *J. Magn. Reson.* **73**, 553–557 (1987).
8. J. C. J. Barna, E. D. Laue, M. R. Mayger, J. Skilling, and S. J. P. Worrall, Exponential sampling, an alternative method for sampling in two-dimensional NMR experiments, *J. Magn. Reson.* **73**, 69–77 (1987).
9. M. Robin, M.-A. Delsuc, E. Guittet, and J.-Y. Lallemand, Optimized acquisition and processing schemes in three-dimensional NMR spectroscopy, *J. Magn. Reson.* **92**, 645–650 (1991).
10. P. Schmieder, A. S. Stern, G. Wagner, and J. C. Hoch, Application of nonlinear sampling schemes to COSY-type spectra, *J. Biomol. NMR* **3**, 569–576 (1993).
11. P. Schmieder, A. S. Stern, G. Wagner, and J. C. Hoch, Improved resolution in triple-resonance spectra by nonlinear sampling in the constant-time domain, *J. Biomol. NMR* **4**, 483–490 (1994).
12. J. Skilling and R. K. Bryan, Maximum entropy image reconstruction: general algorithm, *Mon. Not. R. Astron. Soc.* **211**, 111–124 (1984).
13. J. C. Hoch and A. S. Stern, "NMR Data Processing," Chapter 5, Wiley-Liss, New York (1996).
14. P. Schmieder, A. S. Stern, G. Wagner, and J. C. Hoch, Quantification of maximum-entropy spectrum reconstructions, *J. Magn. Reson.* **125**, 332–339 (1997).
15. A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manjekar, and V. Sunderam, "PVM: Parallel Virtual Machine. A Users' Guide and Tutorial for Networked Parallel Computing," MIT Press, Cambridge, Massachusetts (1994). [The software is available at <http://www.epm.ornl.gov/pvm/>.]