December 13, 1994

Parallel PET Reconstruction by EM Iteration with Reduced Processor Communications

Soeren P. Olesen MS, Jens Gregor PHD, Michael G. Thomason PHD

Department of Computer Science

University of Tennessee

Knoxville, TN 37996

Gary T. Smith MD

Department of Radiology

University of Tennessee Medical Center

Knoxville, TN 37920

Technical Report CS-94-256.

Correspondence to:
Michael G. Thomason, PhD
Department of Computer Science
University of Tennessee
107 Ayres Hall
Knoxville, TN 37996–1301
Phone: (615) 974–5067
e-mail: *thomason@cs.utk.edu*

Abstract

PET reconstruction by the EM algorithm is an iterative computation of Poisson emission rates to maximize a likelihood function. The method is time-consuming and, for real scanner data, requires large numerical arrays. To speed up the computation on multiple processors which have their own local memory and communicate by passing messages on a network, a parallel method has been implemented in which processors compute several iterations before exchanging their latest data with other processors. This method is convenient for iterative reconstruction using a relatively small number of standard workstations on a local area network, i.e., for implementation on computer resources commonly available in clinical and research environments and for which reducing communication among processors is desirable. Computational aspects of the method are explained and illustrated with 2-D reconstructions from a simulation and from sinograms produced by a PET scanner. 512 iterations are computed on a local area network of workstations and, for reference, on a distributed-memory multiprocessor computer as well. The method is capable of producing high quality reconstructions with significant speed-up.

In emission tomography reconstruction by spatial statistical methods (cf., [1, 2]), algorithms for maximum likelihood estimation of parameters tend to be iterations with long computation times involving large arrays. One way to address these problems of time and memory is to implement the iteration on multiple processors in parallel, for example, using a massively parallel computer [3, 4] or taking advantage of specific interconnection topologies [5, 6].

Parallel iteration can be distributed over a network of workstations. Distributed iteration generally reduces the memory required at any one processor by allowing each processor to use only portions of large, sparse matrices. Distributed iteration on multiple processors may also reduce the average time to compute one iteration; however, communication among processors is time-consuming on typical networks, so an objective here is to reduce communications substantially while still producing satisfactory reconstructions.

This paper describes a method for distributed-memory, iterative reconstruction by the EM algorithm using relatively few processors with reduced communications. The approach is motivated by (*i*) the potential for reducing processor communications without compromising fundamental algorithmic requirements such as convergence [7], and (*ii*) the availability of distributed-memory resources not only as multiprocessor computers [8] but also as local area networks of standard workstations [9]. Empirical results are given for three sinograms: a simulated phantom, a 2-fluoro-2-deoxyglucose (FDG) brain scan, and a ^{13}N-ammonia cardiac scan. 512 iterations are computed. Results in convergence, percentage error, and timing are reported for reconstruction on a conventional network of workstations and, for comparison, on a multiprocessor computer.

Aspects of EM-ML Iterative Reconstruction

The EM-ML (expectation-maximization, maximum-likelihood) algorithm [1, 2, 10, 11, 12] for estimating Poisson parameters $\{\lambda_b \mid 1 \leq b \leq B\}$ at B emission sites, given the non-negative integer counts $\{n_d^* \mid 1 \leq d \leq D\}$ for D detector-pairs or tubes, yields the same form of *multiplicative iteration* for each pixel b, namely,

$$\begin{aligned} \lambda_b^{k+1} &= \lambda_b^k \sum_d \frac{n_d^* p_{bd}}{\lambda_d^{*k}} \\ &= \lambda_b^k M_b^k \end{aligned}$$

where p_{bd} is the probability that emission from site b is detected in tube d;

$$\lambda_d^{*k} = \sum_b \lambda_b^k p_{bd};$$

and

$$M_b^k = \sum_d \frac{n_d^* p_{bd}}{\lambda_d^{*k}}$$

is the multiplier computed for pixel b at iteration k. The pixels in the k^{th} iteration are written as the vector $\boldsymbol{\lambda}^k = (\lambda_1^k, \lambda_2^k, \ldots, \lambda_B^k)$ in B-space (*pixel space*). The vector $\boldsymbol{\lambda}^{*k} = (\lambda_1^{*k}, \ldots, \lambda_D^{*k})$ is the projection of $\boldsymbol{\lambda}^k$ into D-space (*detector space*) via the matrix $P = [p_{bd}]$. Proofs given elsewhere (cf., [12]) show that pixel vector $\boldsymbol{\lambda}^k$ converges to a fixed point vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_B)$ which is a maximum likelihood solution. One aspect is that the *Kullback discrimination measure*

$$D(n^*, \lambda^{*k}) = \sum_d n_d^* \log \frac{n_d^*}{\lambda_d^{*k}}$$

converges monotonically as $\lambda_d^{*k} \to n_d^*$ for each d. Computation of $D(n^*, \lambda^{*k})$ assumes positive values $(n_d^*, \lambda_d^{*k} > 0)$ and count-normalization $(\sum_d n_d^* = \sum_d \lambda_d^{*k})$. In practice, inconsistencies due to factors such as random counts, scatter, and numerical inaccuracies mean that differences between n_d^* and λ_d^{*k} may remain detectable for some d and the Kullback measure may approach a positive value.

Any index d such that $n_d^* = 0$ or b such that λ_b^k is underflowing to 0 is eliminated from the computation, assuring that multipliers $\{M_b^k\}$ remain greater than 0 throughout the iteration. The EM-ML vectors remain normalized in counts, i.e., the initial image vector $\boldsymbol{\lambda}^0$ is specified to have

$$\sum_b \lambda_b^0 = \sum_d n_d^*,$$

and the normalization

$$\sum_b \lambda_b^k = \sum_d \lambda_d^{*k} = \sum_d n_d^*$$

is automatically maintained for $k = 0, 1, 2, \ldots$. Reconstruction is initialized with an image $\boldsymbol{\lambda}^0$ in which each pixel λ_b^0 is positive. Iterating then gives

$$\begin{aligned}\lambda_b^{k+1} &= \lambda_b^k M_b^k \\ &= \lambda_b^0 \left(M_b^0 M_b^1 \cdots M_b^k\right),\end{aligned}$$

which shows that pixel λ_b^{k+1} converges to a fixed, positive value λ_b if and only if the multiplier product $M_b^0 \cdots M_b^k$ also converges to a fixed, positive value; this in turn requires that the computed multiplier M_b^k itself converge to 1 [13].

Materials and Methods

Initial Image, Sinograms, and P Matrix

In this study, initial image $\boldsymbol{\lambda}^0$ is a uniform image in which each pixel has the same positive value. 512 iterations are carried out for EM-ML without employing regularization, filtering, or prior probabilities (cf., [14, 15, 16, 17, 18, 19, 20] for some of these methods in PET or SPECT). No explicit terms for scatter or random counts are included in these computations.

Scan data was obtained on an ECAT 921 PET scanner (CTI/Siemens, Knoxville, TN) at the University of Tennessee Medical Center, Knoxville (UTMCK). 2-D reconstructions are 128×128 pixels. The data $\mathbf{n}^* = (n_1^*, n_2^*, \ldots, n_D^*)$ for D detector-pairs or tubes is stored as a sinogram with 192 rows and 160 columns. The P matrix for the EM algorithm has 128×128 rows and 192×160 columns. Entry p_{bd} is a double-precision, floating-point value computed as the area of intersection of tube d with pixel b (approximated by its inscribed circle) relative to the sum-total intersection of that pixel with all tubes. This yields approximately 8×10^6 nonzero entries in P.

Three sinograms are used. The first is a 192×160 sinogram obtained by forward projecting a version of the Shepp-Vardi simulated phantom [1] with 128×128 pixels through the P matrix. The second and third are ECAT 921 sinograms produced by FDG brain scan and ^{13}N-ammonia cardiac scan respectively. Each ECAT 921 sinogram is normalized for detector variability and corrected for attenuation using data from a transmission scan obtained prior to radiopharmaceutical injection. Each patient scan follows routine clinical protocols and data processing approved by the UTMCK Institutional Review Board.

Computer Resources

Reconstruction is implemented on a network of workstations and on a multiprocessor computer in the UT-K Computer Science Department.

The multiprocessor computer is a CM-5 (TMC, Cambridge, MA [8]) with 32 nodes which communicate over high-speed, internal networks. The code in C for PET reconstruction uses the CM-5 as a 2^n-node MIMD (multiple-instruction, multiple-data) machine with a RISC processor at each node. Communication is by passing arrays of data among processors.

The code in C for PET reconstruction on a network of standard workstations is essentially the same as the CM-5 code. A virtual 2^n-node MIMD machine is composed of Sun Sparc5 workstations (Sun Microsystems, Mountain View, CA) connected by Ethernet. This implementation uses PVM (Parallel Virtual Machine [9, 21]), version 3.3.3, a general purpose software package that facilitates distributed computation on a network of Unix computers.

Each CM-5 node and Sparc5 workstation has 32 Mbytes local memory. The CM-5 RISC processor has a 32 Mhz clock and a peak execution rate of 4 Mflops. The Sparc5 processor has a 70 Mhz clock and 13.1 Mflops peak rate, but the effective rate of Sparc5 communication on the Ethernet averages at least two orders of magnitude slower than CM-5 inter-processor communication.

Distributed EM-ML Iteration

The implementation of distributed iteration is *block-parallel* [7] with balanced partitioning, meaning that the B-space vector is partitioned into equal-sized, disjoint subsets of pixels assigned to different processors for iteration. For convenience in manipulating the P matrix [22], the blocks are contiguous rows of pixels in the 2-D image. Each processor stores only those p_{bd}'s that are non-zero for its pixel block, and each processor has approximately the same number of non-zero p_{bd}'s to deal with.

Full synchronization means that all processors communicate data after every iteration; this is a parallel version of the conventional, single-processor computation. *Reduced synchronization* means that each processor follows the same fixed schedule of computing several iterations of its block of pixels, then communicating its latest data. In either case, when a synchronization occurs, each processor communicates its current contribution to entries in the D-space vector $\boldsymbol{\lambda}^{*k}$, i.e., processor $\mathcal{P}_i$ communicates the vector $(\lambda_{i1}^{*k}, \ldots, \lambda_{iD}^{*k})$ where

$$\lambda_{id}^{*k} = \sum_{b_i} \lambda_{b_i}^{k} p_{b_i d}$$

and the sum is over pixel indices $\{b_i\}$ in the block assigned to processor $\mathcal{P}_i$ for iteration. Each processor $\mathcal{P}_i$ updates its own block of pixels and its contributions to the D-space vector at every iteration. In doing so, processor $\mathcal{P}_i$ computes its pixel multipliers $\{M_{b_i}^k\}$ using a combination of its block's current values and the D-space values contributed by the other processors at the last synchronization.

Unlike fully synchronized EM-ML computation, distributed iteration with reduced synchronization does not automatically maintain normalization of counts for the pixel vector $\boldsymbol{\lambda}^k$. Normalization is not automatically recovered and convergence of multipliers is adversely affected. These characteristics are controlled at synchronization times and between synchronizations in the distributed algorithm in the following two ways.

(1). At each synchronization, each processor obtains the up-to-date D-space vector $\boldsymbol{\lambda}^{*k} = (\lambda_1^{*k}, \ldots, \lambda_D^{*k})$ where

$$\lambda_d^{*k} = \sum_i \lambda_{id}^{*k}.$$

Although all values in $\boldsymbol{\lambda}^{*k}$ are based on the most recent iteration, the vector may have lost normalization of detector counts since the previous synchronization; therefore, each processor computes the normalization scalar

$$\alpha_D^k = \frac{\sum_d n_d^*}{\sum_d \lambda_d^{*k}}$$

and renormalizes in D-space by multiplying $\boldsymbol{\lambda}^{*k}$ by this scalar. Each processor similarly renormalizes its block of the B-space vector $\boldsymbol{\lambda}^k$ by the scalar

$$\begin{aligned} \alpha_B^k &= \frac{\sum_d n_d^*}{\sum_b \lambda_b^k} \\ &= \frac{\sum_d n_d^*}{\sum_i \sum_{b_i} \lambda_{b_i}^k} \end{aligned}$$

which is in fact identical to α_D^k because

$$\begin{aligned} \sum_b \lambda_b^k &= \sum_b \lambda_b^k \sum_d p_{bd} \\ &= \sum_d \sum_i \sum_{b_i} \lambda_{b_i}^k p_{b_i d} \\ &= \sum_d \sum_i \lambda_{id}^{*k} \\ &= \sum_d \lambda_d^{*k}. \end{aligned}$$

These renormalizations complete the synchronization phase.

(2). In this multiplicative iterative method, the convergence of pixel λ_b^k to a fixed, positive value λ_b requires that the multiplier M_b^k itself must converge to 1. This means that the range of multiplier values is contained in a finite interval $[L, U]$, $0 < L \leq 1$ and $1 \leq U$, which during iteration has L monotonically nondecreasing to 1 and U monotonically nonincreasing to 1. As a conservative estimate of this interval, each processor $\mathcal{P}_i$ maintains for reference an upper bound U_i and a lower bound L_i. $\mathcal{P}_i$ adjusts these bounds towards 1 over the course of the iteration. At the first computation of multipliers after each synchronization when correct normalization is guaranteed, $\mathcal{P}_i$ checks its computed multipliers to decrease U_i or increase L_i accordingly. U_i is decreased towards 1 when all multipliers computed by $\mathcal{P}_i$ immediately after a synchronization are less than its current value; L_i is increased towards 1 when all computed multipliers exceed its current value. At each iteration, each pixel $\lambda^k_{b_i}$ for which the computed multiplier $M^k_{b_i}$ is in the interval $[L_i, U_i]$ is updated.

Network and Algorithm Parameters

The empirical results concern convergence, percentage error, and time of computation for distributed EM-ML iteration under the following conditions.

Control of Network. All runs on the CM-5 and the workstations are made in dedicated modes which prohibit user processes not associated with the EM-ML computation. Systems traffic on the Ethernet is suppressed as far as possible.

Number of Processors. Results are given for 8 and 16 processors. As the number of processors is reduced, each processor must deal with larger portions of the P matrix; this begins to cause considerable memory paging on processors with 32 Mbytes local memory, which slows the computation severely. As the number of processors is increased, the Ethernet tends to saturate with messages even when synchronizations are reduced, and this also severely impacts the time of computation. The number of processors is a practical compromise between processing power and communication for the workstations, the network, the P matrix, and the sinograms involved in these runs.

Reduced Synchronization. All runs consist of 512 iterations. Large changes in numerous pixels occur in the first few iterations, so full synchronization is maintained for the first 16 steps; thereafter, the number of iterations between synchronizations is linearly increased to a maximum value used for the rest of the run. Reconstructions are computed for caps of 1, 4, and 8 iterations between synchronizations. Cap 1 is full synchronization. Cap 4 results in 140 synchronizations and cap 8 in 80 synchronizations, respectively 27.3% and 15.6% of full synchronization for 512 iterations.

Results

Reconstructed Images

Images in figures 1-4 are displayed as integer pixels of one byte each.

Figure 1 is the simulated phantom $\boldsymbol{\lambda}^{phan}$. Figure 2 shows its reconstructions at iterations 32 and 512 for the three caps. Figure 3 shows the reconstructed brain scan (pixel size 2.275-mm) and figure 4 the reconstructed cardiac scan (pixel size 3.033-mm) for the same parameters. These are reconstructions using 8 processors. Visually, the reconstructions on 16 processors for corresponding parameters are virtually indistinguishable.

Convergence and Percentage Error

Figures 5 and 6 give quantitative measurements of convergence and percentage error in reconstruction. The explicit data points marked in the plots are at iterations that are powers of 2 between 8 and 512. Quantitative measurements are based on the floating-point values computed for pixels.

Figure 5(a) gives plots of the Kullback measure $D(n^*, \lambda^{*k})$ *vs.* iteration number for fully synchronized reconstruction for each of the three sinograms when 8 processors are used. For each sinogram, the values of $D(n^*, \lambda^{*k})$ for reduced synchronizations essentially coincide with those for fully synchronized reconstruction, and the plots for caps 4 and 8 are not distinguishable from the plot for cap 1. Figure 5(b) plots $D(n^*, \lambda^{*k})$ for 16 processors. Convergence trends are the same as figure 5(a).

For 8 processors and the simulated phantom, figure 6(a) gives the percentage error $\mathcal{E}^k$ in the reconstructed pixel vector $\boldsymbol{\lambda}^k$ referenced to the original vector $\boldsymbol{\lambda}^{phan}$, computed as

$$\mathcal{E}^k = \frac{\sum_b (\lambda_b^k - \lambda_b^{phan})^2}{\sum_b (\lambda_b^{phan})^2} \times 100.$$

Differences in $\mathcal{E}^k$ for full and reduced synchronizations at corresponding iterations are negligible. At 512 iterations, all values in figure 6(a) are less than 0.15%. For 16 processors, figure 6(b) shows slightly different values because the finer partitioning of the P matrix among processors induces different patterns of small-scale numerical inaccuracies during distributed iteration. At 512 iterations, all values in figure 6(b) are less than 0.25%.

Fully synchronized reconstructions also serve as references with which to measure errors associated with reduced synchronization. This percentage error is computed for each sinogram, for caps 4 and 8, and for 8 and 16 processors by using the reconstructions for cap 1 at corresponding iterations as the references. For the simulated phantom, this error in all cases is less than 0.02% throughout the iteration; for the brain scan, less than 0.02%; and for the cardiac scan, less than 0.8%.

Time of Computation

The wall-clock time to compute 512 iterations is essentially the same for each of the three sinograms for corresponding settings of network and algorithm parameters. Figure 7 shows these times for the sinogram of the brain scan for 8 and 16 processors on the CM-5 and on the Ethernet-connected workstations. Figure 7 breaks the wall-clock time down into time spent in computing iterations (i.e., in computing and applying the multipliers), in communicating data, and in idling while waiting to initiate synchronizations.

Wall-clock times on the CM-5 show little dependence on the number of synchronizations because communication is fast among its relatively slower but tightly-coupled processors. 16 CM-5 processors require approximately 55% of the time taken by 8 processors.

Wall-clock times for the Sparc5 workstations on the Ethernet are strongly influenced both by the cap and by the number of processors. Although 16 processors take about 55% the time of 8 to compute and use the multiplers, they also increase the network traffic substantially, and the effective rate of communication deteriorates sharply when the Ethernet is saturated with messages. A contributing factor is that PVM version 3.3.3 uses algorithms that are robust but not yet tuned for optimality in communications performance [21].

Communication accounts for about 75%, 50%, and 40% of wall-clock time for 8 Sparc5s with caps 1, 4, and 8 respectively. Comparable numbers for 16 Sparc5s are 85%, 80%, and 72%. A consequence is that 8 workstations require less wall-clock time than 16 workstations for all three caps. Using 8 Sparc5s with iteration cap 8 yields a wall-clock time comparable to 16 nodes on the CM-5, i.e., an average of less than 2 seconds per iteration over the run of 512 iterations.

Conclusions

We conclude that distributed EM-ML iteration with reduced communications is effective in speeding up the reconstruction when network, workstation, and algorithm parameters are selected appropriately. For the sinograms and P matrix in this study, the algorithm for reduced synchronization with caps 4 or 8 distributed over 8 or 16 processors produces reconstructions that are close (visually and quantified by percentage error) to fully synchronized reconstructions. Convergence as measured by Kullback discrimination is also comparable to fully synchronized computation.

Workstation and network characteristics must be taken into account to prevent adverse trends such as excessive processor paging or network saturation with messages. When several hundred iterations are computed, a comparatively slow network (the Ethernet) of 8 standard workstations (Sparc5) using generic distributed software (PVM) attains an average time per iteration approaching clinically acceptable rates.

References

[1] L A Shepp and Y Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Trans. Med. Imaging*, 1:113–122, 1982.

[2] D W Townsend and M Defrise. Image reconstruction methods in positron tomography. Technical report, CERN (93-02), Geneva, Switzerland, 1993.

[3] M I Miller and B Roysam. Bayesian image reconstruction for emission tomography incorporating Good's roughness prior on massively parallel processors. *Proc. Natl. Acad. Sci.*, 88:3223–3227, 1991.

[4] C S Butler and M I Miller. Maximum a posteriori estimation for SPECT using regularization techniques on massively parallel computers. *IEEE Trans. Med. Imaging*, 12:84–89, 1993.

[5] C M Chen, C Y Lee, and Z H Cho. Parallelization of the EM algorithm for 3–D PET image reconstruction. *IEEE Trans. Med. Imaging*, 10:513–522, 1991.

[6] C-M Chen and S-Y Lee. On parallelizing the EM algorithm for PET reconstruction. *IEEE Trans. Parallel Dist. Sys.*, 5:860–873, 1994.

[7] D P Bertsekas and J N Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Englewood Cliffs, NJ, 1989.

[8] Thinking Machines Corporation, Cambridge, MA. *CM–5 Technical Summary*, January 1992.

[9] A Geist, A Beguelin, J Dongarra, W Jiang, R Manchek, and V Sunderam. PVM 3 User's Guide and Reference Manual. Technical report, Oak Ridge National Laboratory (ORNL/TM-12187), Oak Ridge, TN, 1994.

[10] A P Dempster, N M Laird, and D B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc.*, 39:1–38, 1977.

[11] K Lange and R Carson. EM reconstruction algorithms for emission and transmission tomography. *J. Comput. Assist. Tomogr.*, 8:306–316, 1984.

[12] Y Vardi, L A Shepp, and L Kaufman. A statistical model of positron emission tomography. *J. Am. Stat. Assoc.*, 80:8–20, 1985.

[13] K Knopp. *Infinite Sequences and Series.* Dover, New York, 1956.

[14] D L Snyder and M J Miller. The use of sieves to stabilize images produced with the EM algorithm for emission tomography. *IEEE Trans. Nucl. Sci.*, 32:3864–3870, 1985.

[15] E Levitan and G T Herman. A maximum a posteriori probability expectation maximization algorithm for image reconstruction in emission tomography. *IEEE Trans. Med. Imaging*, 6:185–192, 1987.

[16] T Hebert and R Leahy. A generalized EM algorithm for 3–D Bayesian reconstruction from Poisson data using Gibbs priors. *IEEE Trans. Med. Imaging*, 8:194–202, 1989.

[17] A R De Pierro. Multiplicative iterative methods in computed tomography. In *Mathematical Methods in Tomography*, pages 167–186. Springer-Verlag, Berlin, 1991.

[18] V E Johnson, W H Wong, X Hu, and C-T Chen. Bayesian restoration of PET images using Gibbs priors. In *Information Processing in Medical Imaging, IPMI89*, pages 15–28. Wiley-Liss, New York, 1991.

[19] T R Miller and J W Wallis. Clinically important characteristics of maximum-likelihood reconstruction. *J. Nucl. Med.*, 33:1678–1684, 1992.

[20] J W Wallis and T R Miller. Rapidly converging iterative reconstruction algorithms in single-photon emission computed tomography. *J. Nucl. Med.*, 34:1793–1800, 1993.

[21] V S Sunderam, G A Geist, J Dongarra, and R Manchek. The PVM concurrent computing system: evolution, experiences, and trends. *Parallel Computing*, 20:531–547, 1994.

[22] R Barrett, M Berry, T F Chan, J Demmel, J Donato, J Dongarra, V Eijkhout, R Pozo, C Romine, and H van der Vorst. *Templates: Building Blocks for Iterative Methods.* SIAM, Philadelphia, PA, 1993.

Figure Captions

Figure 1. 128×128 pixel simulated phantom.

Figure 2. Reconstructions of phantom with 8 processors. Caps 1, 4, and 8 from left to right. (a)-(c) are 32 iterations; (d)-(f) are 512 iterations.

Figure 3. Reconstructions of brain scan with 8 processors. Organized as figure 2.

Figure 4. Reconstructions of cardiac scan with 8 processors. Organized as figure 2.

Figure 5. Kullback measure $D(n^*, \lambda^{*k})$ using natural logarithms. (a) 8 processors; (b) 16 processors.

Figure 6. Percentage error in phantom reconstructions with respect to original. (a) 8 processors; (b) 16 processors.

Figure 7. Wall-clock time for 512 iterations, broken down into time spent idling (waiting to synchronize), communicating, and computing.

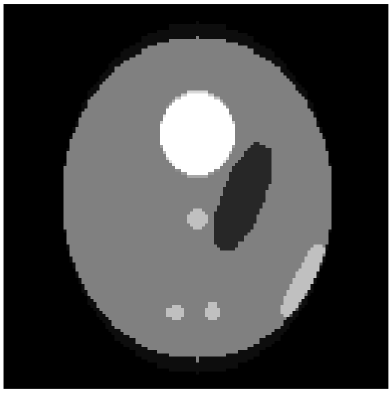
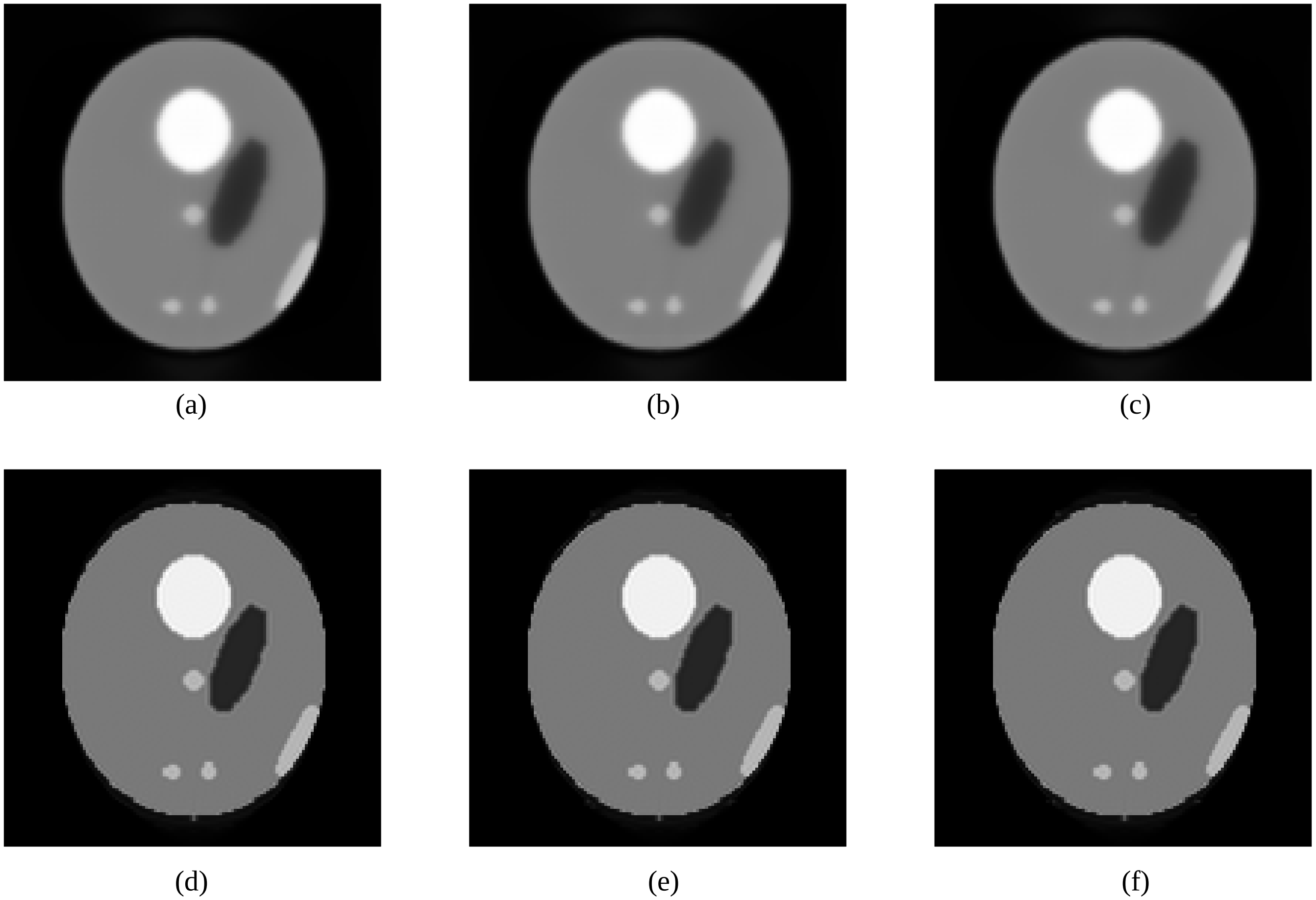


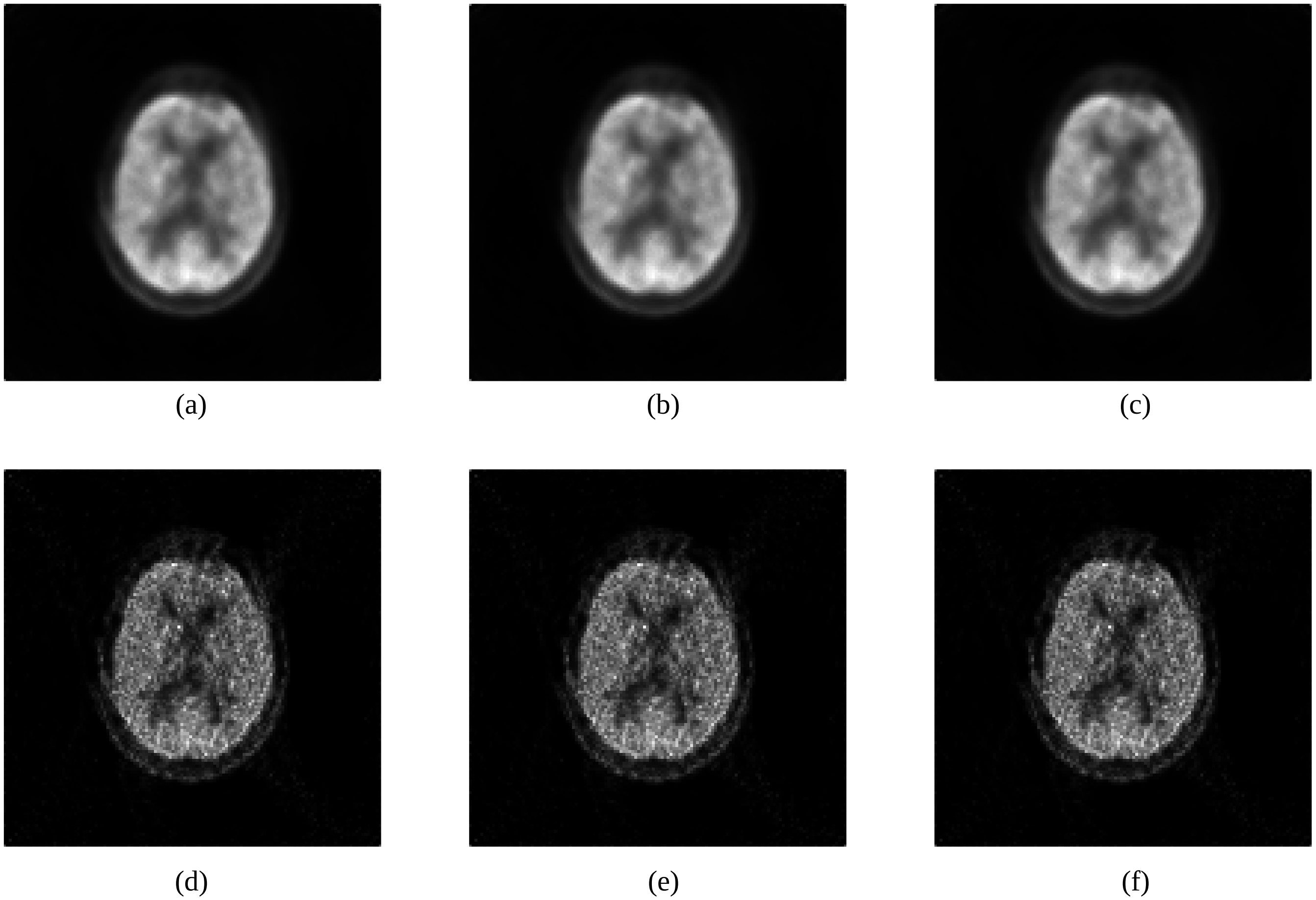
Figure 1.



(a) (b) (c)

(d) (e) (f)

Figure 2.



(a) (b) (c)

(d) (e) (f)

Figure 3.

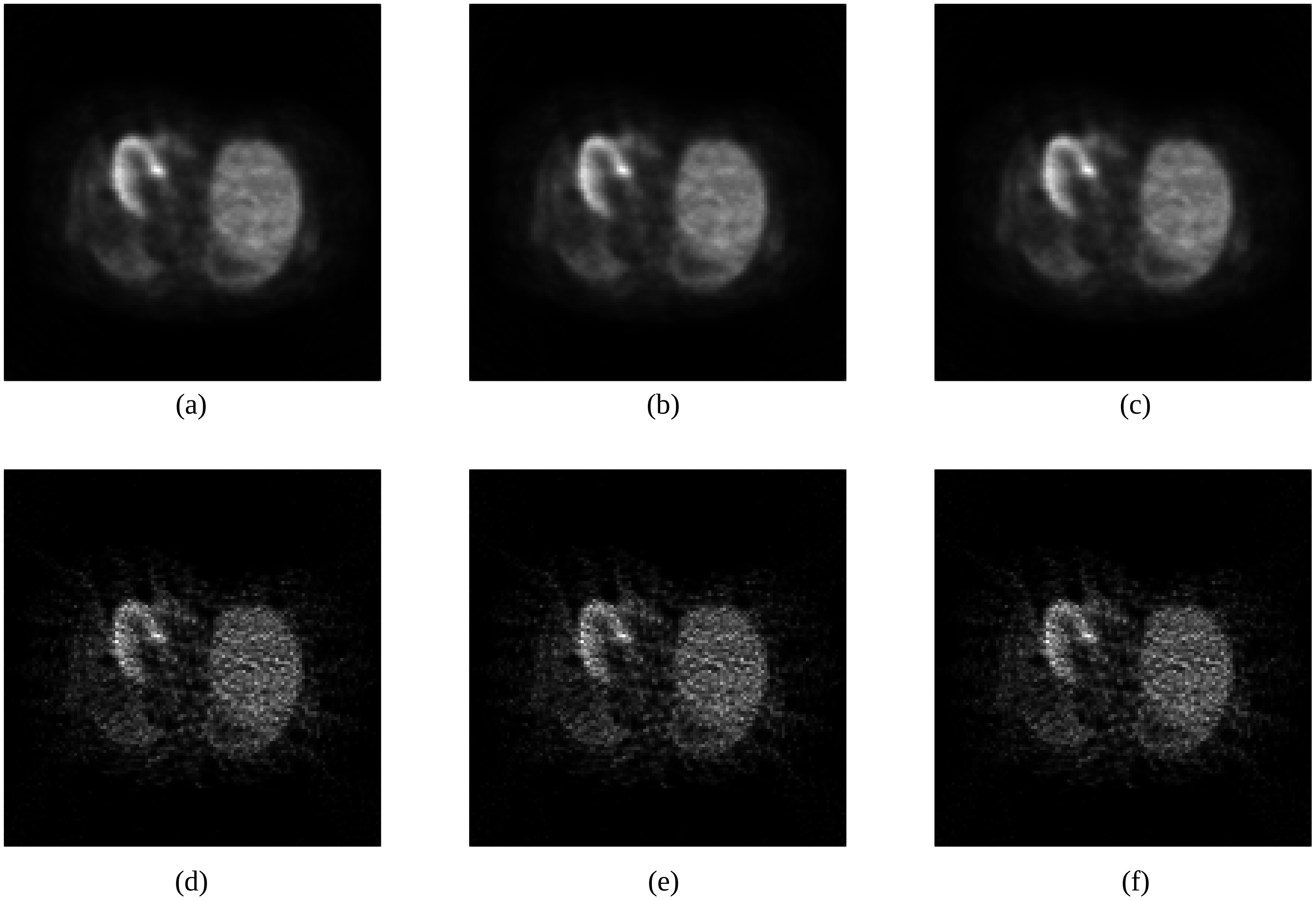


Figure 4.

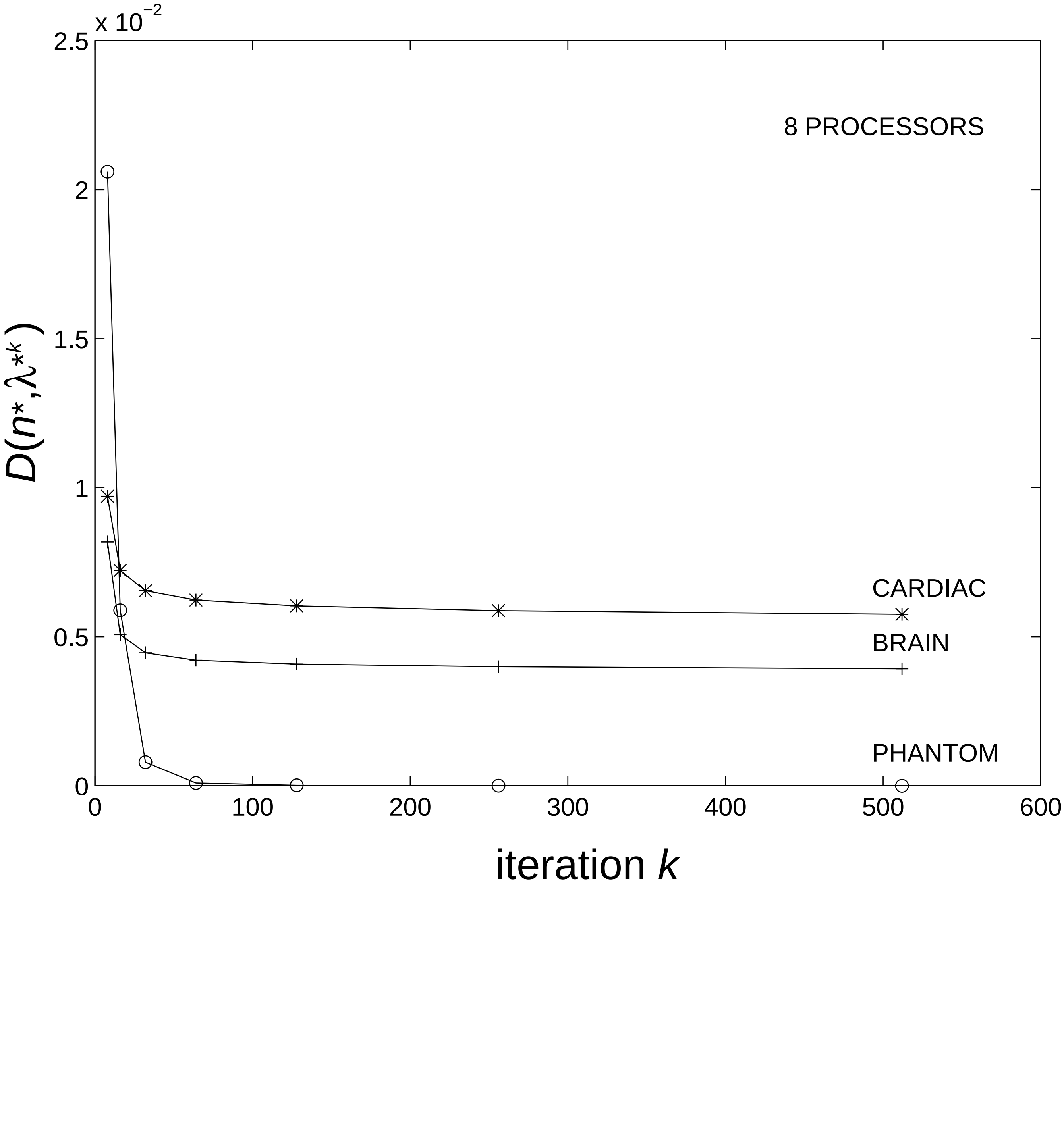

x 10^−2
2.5
2
1.5
1
0.5
0
0
100
200
300
400
500
600
8 PROCESSORS
CARDIAC
BRAIN
PHANTOM
D(n*,λ*^k)
iteration k

Figure 5a.

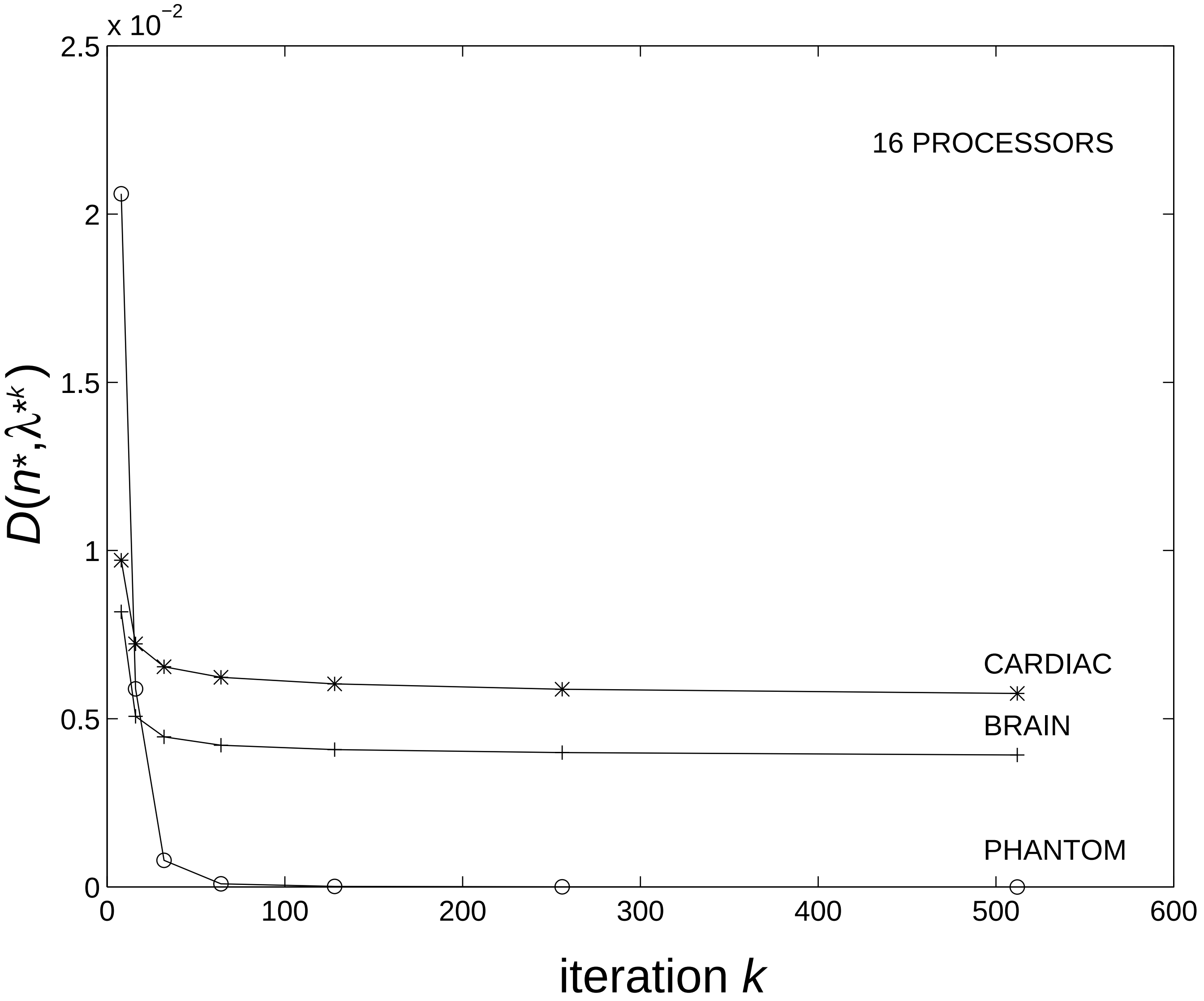

x 10^−2
2.5
2
1.5
1
0.5
0
0
100
200
300
400
500
600
16 PROCESSORS
CARDIAC
BRAIN
PHANTOM
$D(\eta^*,\lambda^{*k})$
iteration k

Figure 5b.

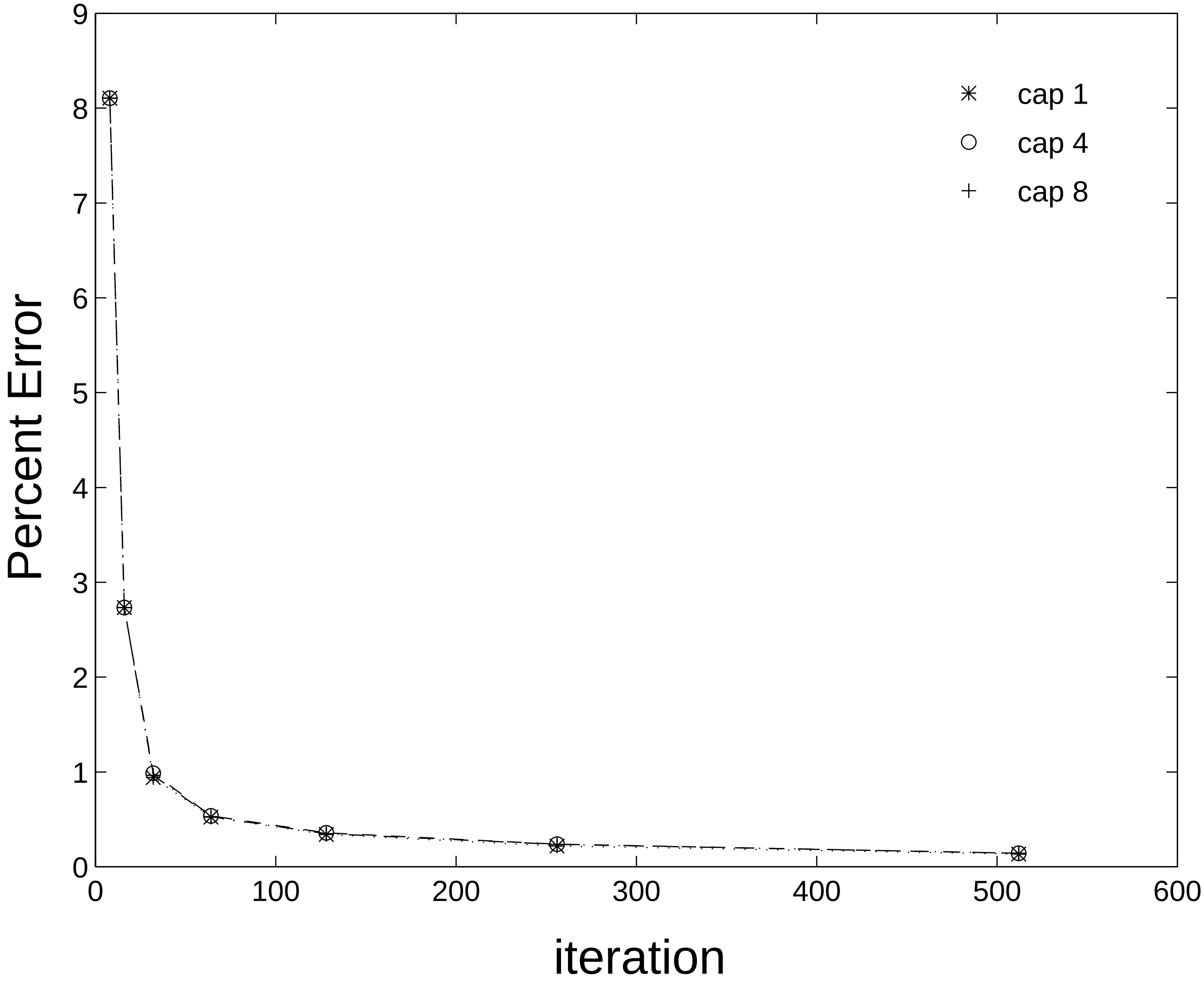

Percent Error
iteration
0
1
2
3
4
5
6
7
8
9
0
100
200
300
400
500
600
✳ cap 1
○ cap 4
+ cap 8

Figure 6a.

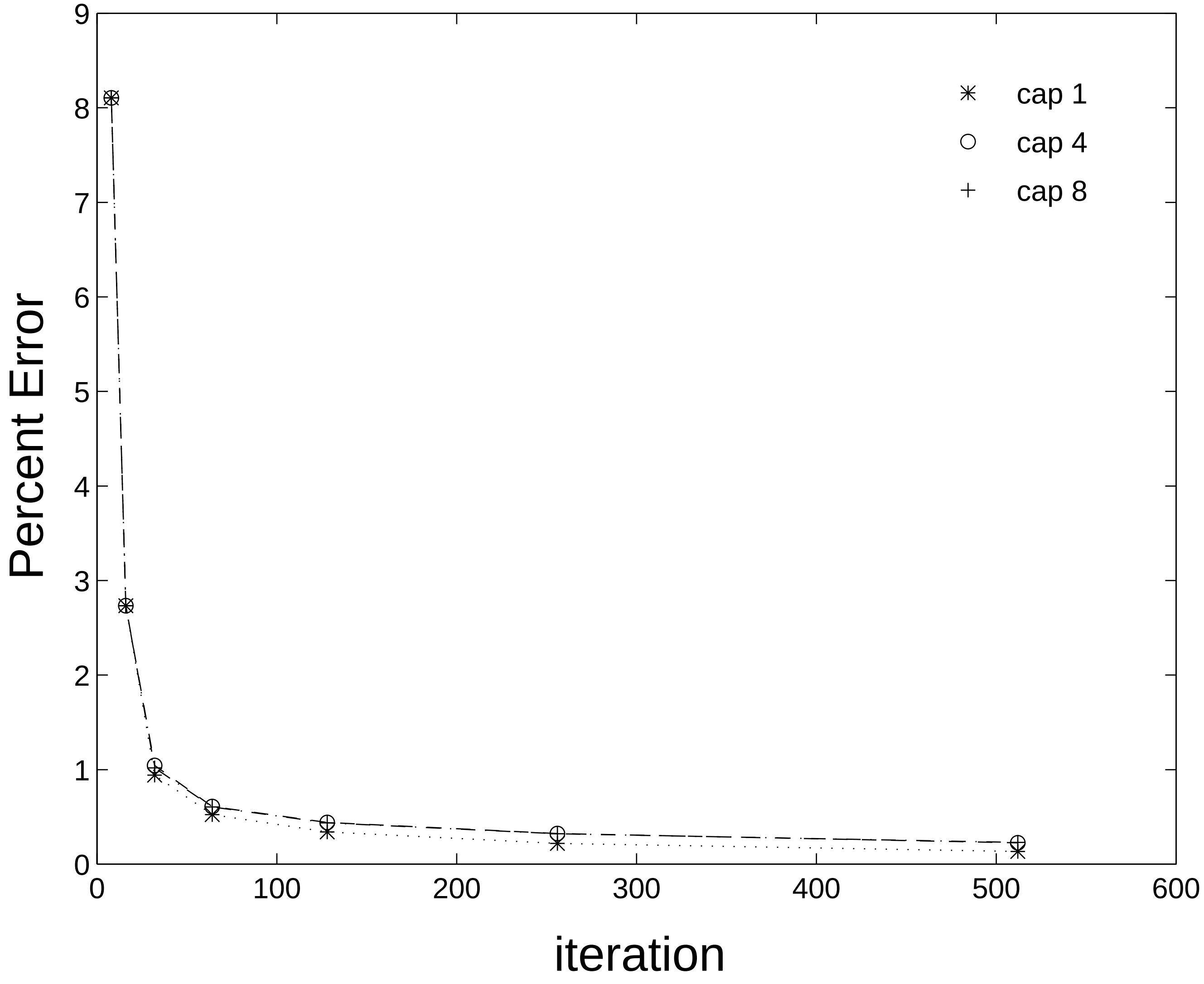

Percent Error
iteration
0
1
2
3
4
5
6
7
8
9
0
100
200
300
400
500
600
✳ cap 1
○ cap 4
+ cap 8

Figure 6b.

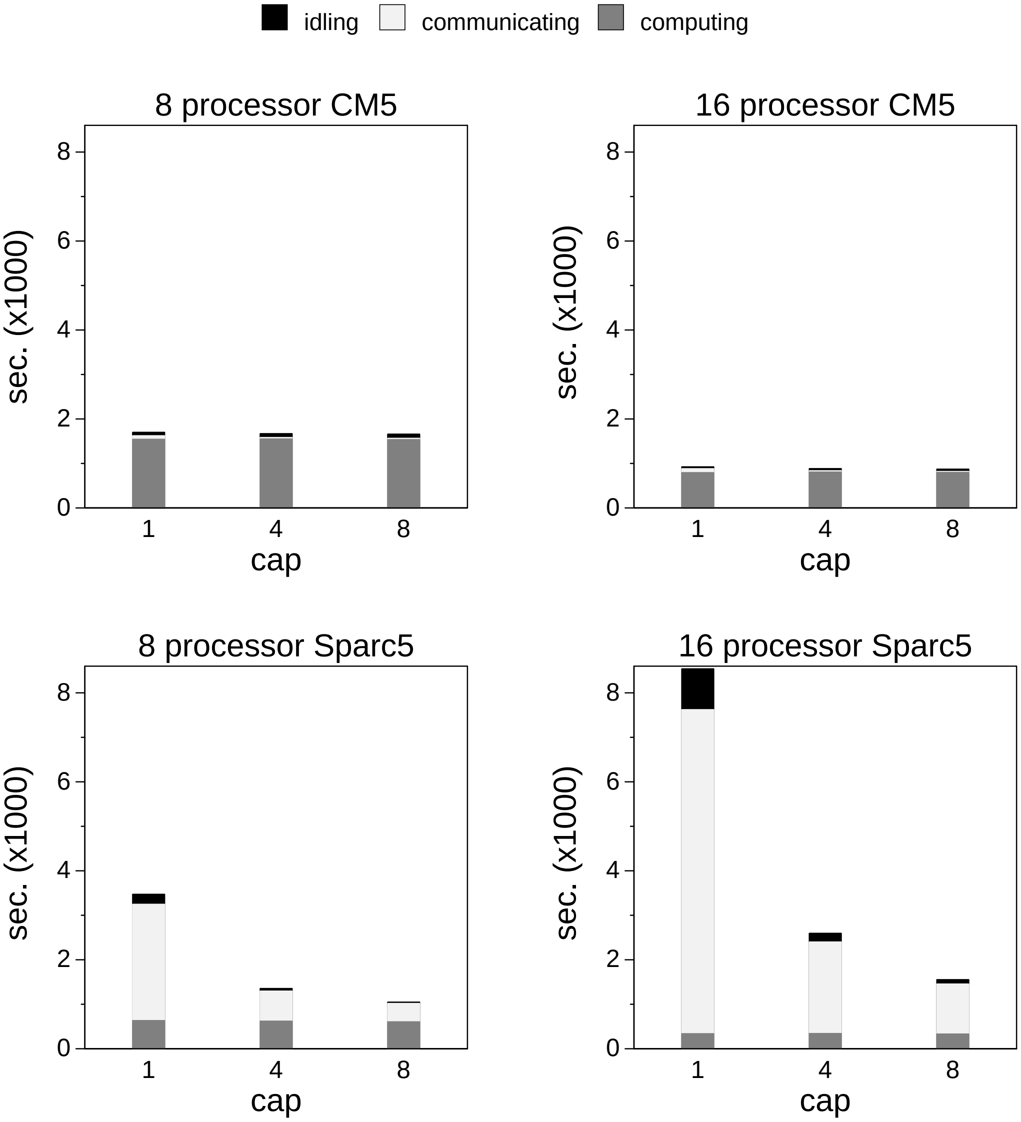

idling
communicating
computing
8 processor CM5
16 processor CM5
8 processor Sparc5
16 processor Sparc5
sec. (x1000)
0
2
4
6
8
cap
1
4
8

Figure 7.