. **LA-UR** -85-3597

CONF-8509201--1

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

NOV 0 0 1985

TITLE: CHAOTIC ITERATION AND PARALLEL DIVERGENCE

LA-UR--85-3597

DE86 002420

AUTHOR(S): Bruce R. Wienke, X-4
Robert E. Hiromoto, C-3

MASTER

SUBMITTED TO: "PARALLEL COMPUTING"
West Berlin, Germany, Sept. 23-26, 1985

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes.

The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

FORM NO 836 R4
ST NO 2629 5/81

CHAOTIC ITERATION AND PARALLEL DIVERGENCE

B. R. Wienke
Applied Theoretical Physics Division
Los Alamos National Laboratory
Los Alamos, N.M. 87545

R. E. Hiromoto
Computing and Communications Division
Los Alamos National Laboratory
Los Alamos, N.M. 87545

Parallelization of standard multigroup methods used to solve the linear (Boltzmann) transport equation in the discrete ordinates (S_n) representation, coupled chaotic iteration schemes, and convergence are the focus of this analysis. On the Denelcor HEP, we investigate the parallel extension of two serial iteration schemes, categorize speedup, and perform some numerical experiments in chaotic iteration.

1. INTRODUCTION

Solutions to the linear transport equation for neutral and charged particles occupy a significant portion of computational efforts.[1,2] We have parallelized two iterative multigroup techniques used to solve the transport equation in the discrete ordinates representation (S_n) and performed some numerical experiments with chaotic iteration. The multigroup S_n picture consists of a step discretization of angular and energy operators and forms the backbone of iterative solution algorithms in many applications. Calculations are performed on the Denelcor HEP[3,4], incorporated into the Los Alamos Integrated Computing Network (ICN) for research purposes. The HEP is a 64-bit MIMD machine consisting of up to 16 process execution modules (PEMs), each capable of executing 64 processes concurrently. Some results of the study are briefly communicated in the following.

2. PHENOMENOLOGY

The multigroup, discrete ordinates equation can be written[5,6]

$$v_g^{-1}\frac{\partial \varphi_{gm}}{\partial t} + \Omega\cdot\nabla\varphi_{gm} + \sigma_g\varphi_{gm} = \sum_{h=1}^{G}\sum_{l=0}^{M-1}\sigma_{hg}^{l}\varphi_h^l P_l(\Omega_m) + Q_{gm}\ , \qquad (1)$$

as a compact representation of the transport equation with φ the particle flux, t the time, Q the external source, v the velocity, σ the transport cross section, and Ω a unit vector in the direction of particle travel. The indices g and m are the discretization indices on the energy and angular domains, respectively,

$$1 \le g \le G \quad (energy)$$

$$1 \le m \le M \quad (angle).$$

The subscript g denotes appropriate group averaged quantities in Eq. (1). The multigroup angular fluxes are functions of particle position, direction, energy, and time satisfying the relationship

$$\int_{\epsilon_g}^{\epsilon_{g+1}} d\epsilon \int d\Omega \varphi(r,\Omega,\epsilon,t) = 4\pi \sum_{n=1}^{M} w_n \varphi_{gn} , \tag{2}$$

with r the position coordinate (one dimensional slab, cylindrical, or spherical geometry), ϵ the energy and w_n Gaussian quadrature weights for Ω_n which integrate the azimuthally symmetric angular flux. The two quantities with superscripts in Eq. (1) are defined in a Legendre expansion

$$\varphi_h^l = \frac{(2l+1)}{4\pi} \int_{\epsilon_h}^{\epsilon_{h+1}} d\epsilon \int d\Omega \varphi P_l(\Omega) , \tag{3}$$

$$\sigma_{hg}^l = \int_{\epsilon_g}^{\epsilon_{g+1}} d\epsilon \int_{\epsilon_h}^{\epsilon_{h+1}} d\epsilon' \int d\Omega \sigma P_l(\Omega) , \tag{4}$$

with μ the cosine of the angle between the position vector $\mathbf{r}$ and the direction of particle travel $\mathbf{\Omega}$ in the appropriate geometry.

Equation (1) is an integro-differential expression that is solved by successive iteration of the angular flux occurring in the first term of the right hand side.[7-10] Iterations on Eq. (1) with $h=g$ are the within-group, or *inner* iterations, while iterations for $h \neq g$ represent *outer* iterations. It is convenient to rewrite the transport equation in compact operator form, where $\boldsymbol{\Phi}$ is the flux

$$(\mathbf{L} + \mathbf{\Sigma})\boldsymbol{\Phi} = (\mathbf{S} + \mathbf{U} + \mathbf{D})\boldsymbol{\Phi} + \mathbf{Q} , \tag{5}$$

and $\mathbf{L}$, $\mathbf{\Sigma}$, $\mathbf{S}$, $\mathbf{U}$, $\mathbf{D}$ and $\mathbf{Q}$ are the streaming, collision, self-scatter, upscatter, downscatter, and external source operators, respectively. Upscatter or downscatter underscore energy transfers between higher or lower energy groups, while self-scatter implies no energy transfer. Upscatter and downscatter couple different energy groups (*outer*), while self-scatter only couples within-group fluxes (*inner*).

The actual speedup gained by parallel processing an iteration algorithm depends upon a number of factors such as number of processes created, degree to which the algorithm supports parallelization, process communication, and synchronization. Denoting the number of processes as p, the degree to which the algorithm supports parallel implementation as γ, and process instruction and communication overhead as υ, the *theoretical* speedup S is given by[11]

$$S(\gamma,p,\upsilon) = \frac{1}{(1-\gamma) + \gamma/p + \upsilon} . \tag{6}$$

For a completely parallelized algorithm ($\gamma = 1$), the speedup is less than the number of processes when $\upsilon > 0$, that is, $S = p/(1 + p\upsilon)$, and overall gain is limited by process overhead. Under optimal conditions, we expect speedup near the number of processes assigned to the iteration task, but the actual speedup can be markedly suppressed for the above reasons. Efficiency, E, is the ratio of speedup to numbers of processes. Our *actual* speedup is computed by dividing the serial execution time by the parallel execution time. Efficiency is then actual speedup per unit process.

We consider two parallel iteration schemes,[12] termed TPMG (multigroup), and TPCC (chaotic convergence), that take the operational forms (respectively) recalling Eq. (5)

$$(\mathbf{L} + \mathbf{\Sigma})\boldsymbol{\Phi}^{i+1,j+1} = \mathbf{S}\boldsymbol{\Phi}^{i+1,j} + (\mathbf{D} + \mathbf{U})\boldsymbol{\Phi}^{i} + \mathbf{Q} , \tag{7}$$

$$(\mathbf{L} + \mathbf{\Sigma})\boldsymbol{\Phi}^{i+1,j+1} = \mathbf{S}\boldsymbol{\Phi}^{i+1,j} + (1-\xi)(\mathbf{D} + \mathbf{U})\boldsymbol{\Phi}^{i} + \xi(\mathbf{D} + \mathbf{U})\boldsymbol{\Phi}^{i+1,j} + \mathbf{Q} . \tag{8}$$

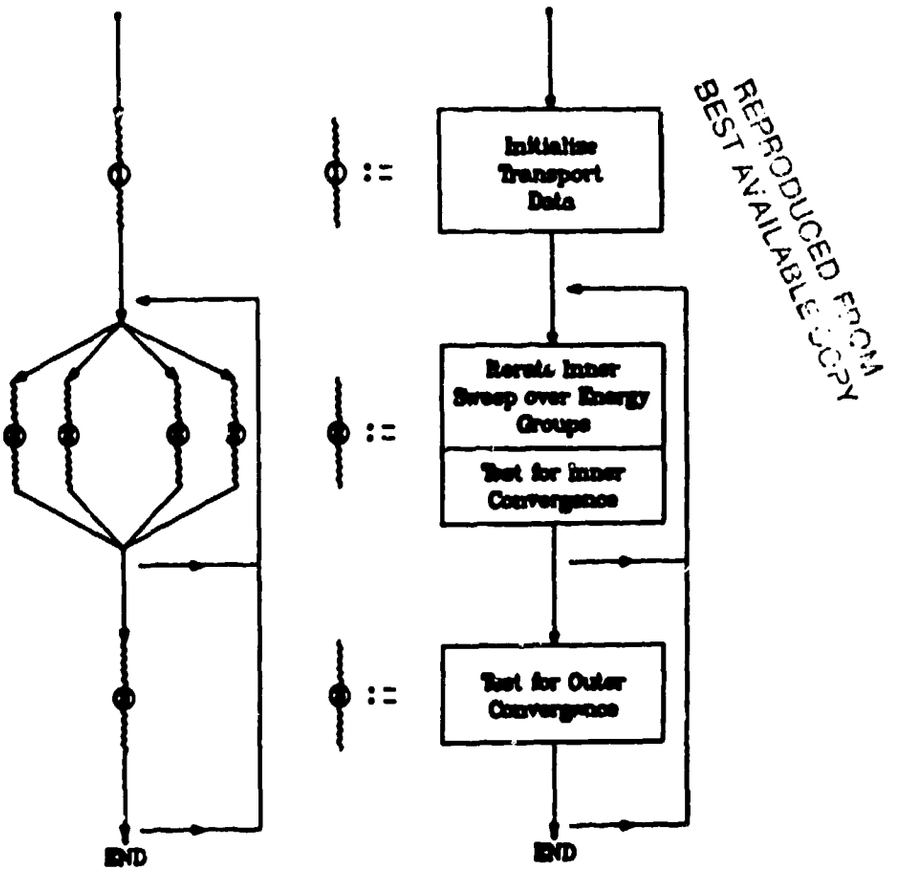
The indices i and j denote the *outer* and *inner* iteration cycles in the the nested scheme used to evaluate Eq. (5). The scheme TPMG is Jacobian-like while TPCC more closely approximates the Gauss-Seidel technique, since the outer source, $(\mathbf{D} + \mathbf{U})\boldsymbol{\Phi}^{i+1,j}$, computed using flux iterates from currently active energy groups, is updated on each inner pass in TPCC, while $(\mathbf{D} + \mathbf{U})\boldsymbol{\Phi}^{i}$, employing flux iterates from the previous cycle, is updated on each outer pass in TPMG. Denoting the number of energy groups as G and processes p, as before, the *chaoticity*, ξ, of a non-ordered parallel algorithm is simply

defined by

$$\xi = \frac{p-1}{G-1} \tag{9}$$

with $1 \leq p \leq G$. Chaoticity of TPCC thus ranges from non-chaotic to fully chaotic as ξ ranges from 0 to 1, while the scheme TPMG remains ordered, as in the serial case. Figure 1 schematizes the parallel flow of TPMG and TPCC, with single sychronization barrier at the inner convergence test.

Figure 1. Parallel Flow And Sychronization Of TPMG And TPCC



REPRODUCED FROM BEST AVAILABLE COPY

3. PARALLEL PERFORMANCE

A representative 16-group, absorber-scatterer, boundary source problem was chosen for study. All groups are coupled and the scattering ratio $\sigma_s/\sigma_s+\sigma_a$, with σ_s and σ_a the scattering and absorption cross sections, varies between 0.66 and 0.88. The scattering (iteration) matrix is thus full, and the spectral radius is bounded by 0.88 (approximately). The test problem was run serially and in parallel. Table 1 contrasts execution time, relative speedup over serial processing time, and iteration count for both schemes. Execution times for TPCC are lower than TPMG for any number of processes assigned to the inner loop, except in the case of 8 or 9 processes for which execution times are roughly the same. Both iteration schemes exhibit their optimal performance with 16 processes. The overall parallel gain with increasing numbers of processes is a monotonic curve for both schemes. Although TPCC is faster than TPMG overall, the relative speedup over single process execution time is greater for the ordered scheme. Both schemes show dramatic relative speedup for 16 processes, and notable execution time minima.

Table 1. Parallel Performance Of TPMG And TPCC

	TPMG	TPMG	TPMG	TPCC	TPCC	TPCC
processes	time (sec)	speedup	iterations	time (sec)	speedup	iterations
1	3235	1.00	312	2054	1.00	198
2	1610	1.91	312	1321	1.56	244
3	1203	2.69	312	1021	2.01	271
4	889	3.64	312	731	2.81	256
5	748	4.33	312	591	3.47	256
6	664	4.87	312	608	3.38	301
7	608	5.32	312	502	4.09	276
8	471	6.86	312	446	4.61	277
9	475	6.81	312	447	4.60	304
10	483	6.70	312	406	5.06	279
11	475	6.81	312	386	5.33	274
12	470	6.80	312	324	6.73	259
13	486	6.45	312	330	6.23	257
14	477	6.78	312	334	6.20	249
15	492	6.58	312	332	6.10	249
16	358	9.07	312	197	10.41	153

While the speedup and efficiency discussed above are useful quantities for describing the relative serial-to-parallel performance of a method, another measure of efficiency, the computation rate (*iterations per second per process*), quantifies the effects of various overheads in a multiprocessing strategy. For $p = 16$, TPMG attains rates of 0.093 and 0.055, while TPCC attains rates of 0.098 and 0.046 for serial and parallel computations. Both strategies suffer a reduction in computation rate due to communication overhead, data conflict, and process straggling in the pipeline. In the present case, TPMG sees a drop of 44% while TPCC suffers a 53% loss in computation rate. Most of this rate reduction can be attributed to architectural limitation on the speedup for 16 processes. The 8 instruction execution pipeline of a single PEM frequently requires only 10 to 12 processes to keep the pipeline operating at its maximum. Analysis of the TPMG and TPCC serial-to-parallel speedups suggests effective pipelines of 9 and 10.4, respectively. The relatively longer pipeline length of TPCC indicates the presence of greater memory traffic limiting efficiency. Should contention be eliminated or minimized, there exists potential for even greater improvement in the TPCC strategy.

4. CONVERGENCE AND ERROR

Relative convergence rates of iterative solution sequences to Eqs. (7) and (8) depend on the magnitude of the spectral radius, κ, with the fastest convergence rates supported by smallest values of the spectral radius and vice-versa. Numerical accuracy depends on the iteration-cutoff convergence parameter, δ, the absolute error, $\mathbf{e}$, and the spectral radius, κ, of the iteration operator in coupled fashion Equation (5) can be rewritten,

$$\boldsymbol{\Phi} = \mathbf{K}\boldsymbol{\Phi} + \mathbf{R} \ , \tag{10}$$

with $\mathbf{K} = (\mathbf{L} + \boldsymbol{\Sigma})^{-1}(\mathbf{S} + \mathbf{D} + \mathbf{U})$ and $\mathbf{R} = (\mathbf{L} + \boldsymbol{\Sigma})^{-1}\mathbf{Q}$. Defining the absolute error after n iterations, $\mathbf{e}^n = (\boldsymbol{\Phi} - \boldsymbol{\Phi}^n)$, with $\boldsymbol{\Phi}$ the exact solution to Eq. (5), we have,

$$\mathbf{e}^n = \mathbf{K}\mathbf{e}^{n-1} = \sum_{k=n+1}^{\infty} (\mathbf{e}^{k-1} - \mathbf{e}^k) \ . \tag{11}$$

The relative convergence criteria, δ, applied to Eq. (5) requires,

$$\left| \Phi^n - \Phi^{n-1} \right| = \left| e^{n-1} - e^n \right| \leq \delta \ . \tag{12}$$

Denoting the spectral radius of **K** as κ, and iterating the middle term of Eq. (12) j times yields,

$$\left| e^{j-1} - e^j \right| \leq \Phi^0 \kappa^j \ , \tag{13}$$

for Φ^0 the initial flux guess, and $0 \leq \kappa < 1$. Therefore, since $\delta \approx \Phi^0 \kappa^n$ from Eqs. (12) and (13),

$$\left| e^n \right| \leq \sum_{k=n+1}^{\infty} \Phi^0 \kappa^k = \delta\kappa \left(1 + \kappa + \kappa^2 + \cdots\right) = \frac{\delta\kappa}{1-\kappa} \ , \tag{14}$$

Relative convergence criteria of 0.001 produce less than 1% error for spectral radii less than 0.9.

5. DIVERGENCE

Although we have had surprisingly good success with the performance of our chaotic iteration scheme, the predictability of general convergence is not straight forward. While it has been shown[13] that a necessary and sufficient condition for chaotic convergence requires a spectral radius less than 1, it is difficult to guarantee that a chaotic updating procedure will have sufficient access to the influx of new iterates required to insure optimum convergence, or, for that matter, convergence at all. In fact it is quite possible for the development of a fixed pathological memory access pattern whose primary results would be to set up oscillations about the required point of convergence. This divergent behavior has been observed in one particular set of numerical experiments for which the corresponding serial iteration scheme converged. To achieve (faster) convergence of a chaotic iteration scheme, we require then some means of providing each individual iteration process with ample access to at least the j^{th} most recent iterated values. Clearly the accessibility to chaotically updated iterates is affected by a superposition of an asynchronous parallel processing system (characterized by such factors as synchronization overhead, memory access and latency times, machine cycle time, and average memory conflict resolution time) onto a particular chaotic algorithm whose parallel computational granularity also affects the rate at which new iterates are produced and consumed. The complexity of analysis for optimum chaotic algorithms may therefore seem enormous.

Fortunately the natural inner-outer iteration strategy, present in our algorithm, avoids the possibility of divergence. By barriering at the outer loop to test for global convergence, we essentially guarantee that each subsequent inner iteration is no more than j iterates behind. This scheme does not address the question of optimum convergence, but it does provide a pathway for the most current iterates to achieve convergence. Further analysis of information flow, group ordering, and outer source dominance (exhibited, for instance, by the group scattering matrix) will help to answer questions of optimum convergence.

6. SUMMARY

The simulations described herein demonstrate successful application of chaotic iteration methods for solving the multigroup equations with scattering and absorption. Parallel extensions of Jacobi and Gauss-Seidel iteration techniques follow easily in the granular multigroup picture, with nominal recoding effort. It seems clear that the art of choosing and implementing chaotic iteration schemes will require greater knowledge and intuition of both parallel machine architecture and chaotic algorithm under consideration. Although complexities may seem unbounded, the introduction of tested chaotic procedures may expose more parallelism than would have been extracted using more traditional parallel decomposition techniques.

7. REFERENCES

(1) L.L. Carter and E.D. Cashwell, Particle Transport Simulation With The Monte Carlo Method, (ERDA Critical Review Series, Oak Ridge, 1975).

(2) B.G. Carlson and K.D. Lathrop, Computing Methods In Reactor Physics, (Gordon and Breach, New York, 1968).

(3) HEP Fortran Users Guide, (Dénelcor, Inc., Aurora, 1982).

(4) H.F. Jordan, Proc. IEEE 72 (1984), 113.

(5) B.R. Wienke, J. Quant. Spect. Rad. Trans. 28 (1982), 311.

(6) G.I. Bell and S. Glasstone, Nuclear Reactor Theory, (Van Nostrand Reinhold, New York, 1970).

(7) B.G. Carl..n, Nuc. Sci. Eng. 61 (1976), 408.

(8) R.E. Alcouffe, E.W. Larsen, W.F. Miller and B.R. Wienke, Nuc. Sci. Eng. 71 (1979), 111.

(9) W.L. Reed, Nuc. Sci. Eng. 45 (1971), 245.

(10) K.D. Lathrop, J. Comp. Phys. 4 (1969), 475.

(11) B.L. Buzbee, Los Alamos Science, 9 (1983), 71.

(12) B.R. Wienke and R.E. Hiromoto, Nuc. Sci. Eng. 90 (1985), 116.

(13) P. Chazan and W. Miranker, Lin. Alg. Appl. 2 (1969), 190.