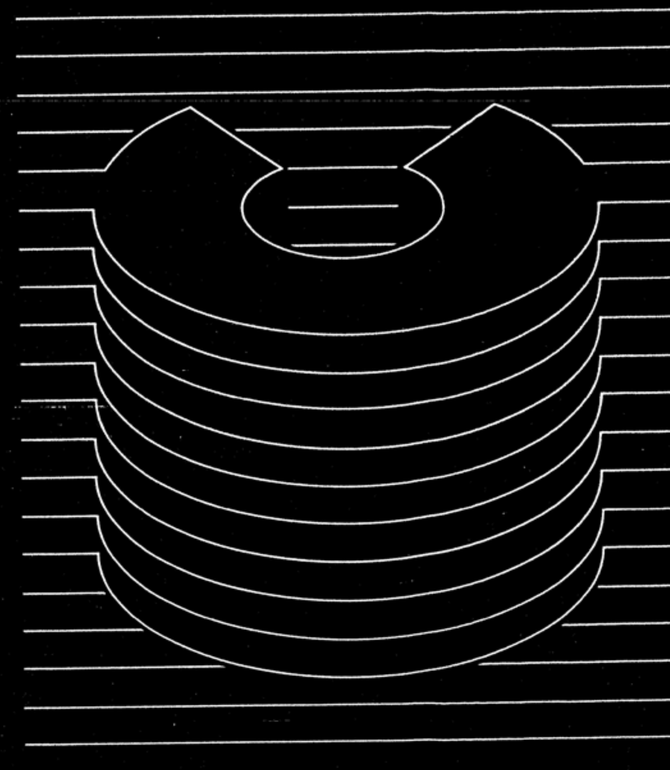
Science and Engineering on Cray Supercomputers

Proceedings of the
Third International Symposium,
Minneapolis, Minnesota, September 1987

Organized by Cray Research, Inc.



CRAY

Science and Engineering on
Cray Supercomputers

A Cray Research, Inc. book

15. Osher S. (1984), Smoothing for Spectral Methods, Chapter 9, Spectral Methods for Partial Differential Equations (Ed. Voigt R. Gottlieb D. and Yousuff Hussaini M.), pp. 209-216, Society for Industrial and Applied Mathematics. Philadelphia.

16. Grischkowsky D. (1970), Self-Focusing of Light by Potassium Vapor, Physical Review Letters, Vol. 24, pp. 866-869.

17. Meyer Y. H. (1980), Multiple Conical Emissions From Near Resonant Laser Propagation in Dense Sodium Vapor, Optics Communications, Vol. 34, pp. 439-444.

18. Bréchignac C. Cahuzac Ph. and deBarre A. (1980), Anomalous Off-Axis Emissions on the Resonance Strontium Line, Illuminated by a Quasi-Resonant Pulsed Laser Light, Optics Communications, Vol. 35, pp. 87-91.

19. Skinner C. H. and Kleiber P. D. (1980), Observation of Anomalous Emission From Laser-Excited Barium Vapor, Physical Review A, Vol. 21, pp. 151-156.

20. Burdge G. L. and Lee C. H. (1982), Characterization of Sideband Emission Generated by Near Resonant Radiation in Sodium Vapor, Applied Physics B, Vol. 28, p. 197.

21. Krasinski J. Gauthier D. J. Malcuit M. S. and Boyd R. W. (1985), Two-Photon Conical Emission, Optics Communications, Vol. 54, pp. 241-245.

22. Harter D. J. and Boyd R. W. (1980), Nearly Degenerate Four-Wave Mixing Enhanced by the ac Stark Effect, IEEE Journal of Quantum Electronics, Vol. QE-16, pp. 1126-1131; Boyd R. W. Raymer M. G. Narum P. and Harter D. J. (1981), Four-Wave Parametric Interactions in a Strongly Driven Two-Level System, Physical Review A, Vol. 24, pp. 411-423.

HYBRID SYMBOLIC-PROCEDURAL PROGRAMMING METHODOLOGY FOR MONTE CARLO PARTICLE TRANSPORT SIMULATIONS

M. Ragheb and D. Gvillo

Department of Nuclear Engineering and Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Ilinois 61801, USA.

ABSTRACT

A hybrid symbolic-procedural programming adaptive methodology on a Cray supercomputer is described. The Monte Carlo simulation of particle transport is considered. An algorithm for variance reduction that is partly heuristic and partly stochastic is analysed. The simulation is conducted in Fortran, whereas the variance reduction process is conducted in the Lisp language. Because the optimization process for the variance reduction parameters lends itself to the use of heuristics, it is useful to use a production-rule analysis system to control the process while running on the Cray. The general-purpose inference engine: HAL-1987 was used for this purpose. In this case a Goal-Tree is constructed using the logical steps that an expert user would use to solve the problem, and is translated into a Knowledge-Base encompassing the expertize of the user. The need for multiple job submissions is avoided, and the speed of the Cray for this type of simulations is taken advantage of. Moreover the Production-Rule System for analysis would guide the user in submitting his input data, checks the data for consistency, and accordingly acts as an intelligent user manual.

The performance of the proposed methodology is compared with that of the non-adaptive approach for particle transport simulations using importance sampling and to the exact analytical solutions for the case of a semi-infinite slab for highly, moderately, and lightly scattering media. The comparison is carried out in terms of the efficiency ratio defined as the product of the labor and error ratios.

INTRODUCTION

A new computer implementation approach to solving problems requiring both procedural and symbolic reasoning is proposed. Occasionally certain problems, whether in science, engineering, or business, etc., fail to completely yield to algorithmic analyses. The algorithms devised still require at least the occasional intervention of the human user to apply heuristics, or learned "rules of thumb", to guide and reorient them. Two examples of such analyses are dynamic mesh reconstruction used with finite element analysis applied to deforming bodies, and variance reduction schemes in Monte Carlo particle transport computations.[1] The advent of the Artificial Intelligence (AI) field of Production-Rule Analysis Systems, or "Expert Systems", has allowed computers to exercise heuristic knowledge usually for the purpose of interacting with, and giving advice to, human users. Two successful examples are MYCIN which is used to diagnose bacterial infections of the blood[2], and the PROSPECTOR mineral exploration systems.[3] One of the goals of this research is to adapt the Production-Rule Analysis system to where it can instead provide a control feature to those algorithms in need of heuristic judgement. The advantages of such a symbolic-procedural coupled configuration include the following: One is that the user is freed from having to work with the analysis, continually supplying it with heuristic knowledge by resubmitting and rerunning multiple cases until an optimal result is obtained. No longer slowed down by the relatively slow response time of the human, the analysis can proceed at an accelerated pace on the computer. And, like all expert systems, once the expertise of the "expert" is distilled and placed into the system in the form of a Knowledge Base, it can be applied systematically and repeatedly on demand, and replicated as well as modified as needed.

"The knowledge manipulation on a grand scale of humanlike intelligence planned for the Fifth Generation will require scaling up by several orders of magnitude in hardware and software."[4] What Feigenbaum, et al., is boldly referring to, of course, is the proposed fifth generation computers which are envisioned to be parallel processing supercomputers capable of symbolic manipulation and symbolic reasoning. The availability of a version of Portable Standard Lisp (PSL) on the Cray X-MP supercomputer enables pre-fifth generation efforts. Lisp on this machine has been shown to be 15 to 30 times the speed of Lisp on a VAX 11/780 and 5 to 15 times the speed of a Symbolics 3600 Lisp machine.[5] In this work, as a pre-fifth generation effort, a general-purpose Production Rule analysis system: HAL-1987, has been developed on this supercomputer, and is used for the generation of Model-Based systems.[6] This provides a technique for modelling engineering devices based on the knowledge of their structure and function rather than on human expertise alone. The Production rule system uses backward-chaining and forward-chaining in an induction-deduction antecedent-consequent logic, and is programmed in PSL. HAL-1987 is the Inference Engine used in the symbolic reasoning portion of the coupled system to be proposed in this work.

The symbolic-procedural analysis problem to be explored here is the adaptive process of variance reduction in Monte Carlo methods where the procedural portion of this coupled system is a variance reduction algorithm using particle tracks scaling.[1] The "Monte Carlo method" is a stochastic simulation method for solving deterministic or probabilistic problems.[7] It is generally defined as "...representing the solution of a problem as a parameter of a hypothetical population and using a random sequence of numbers to construct a sample of the population, form which statistical estimates of the parameters can be obtained."[8] Two distinctive features characterize the Monte Carlo method: The first is the simple structure of the computational algorithm, which simply involves the repetition of a numerical experiment N times, then taking an average over the experiments. The second is that the error in the estimates of the quantities of interest satisfies the proportionality: $\varepsilon \propto \sigma/\sqrt{N}$ where σ is the square root of the variance of the sampled process and N is the number of experiments. So that reduced error bounds can be obtained either by increasing the number of experiments, N, or decreasing the variance σ^2 of the statistical sampling process.[7] Obtaining a sample size sufficiently large to produce an adequate number of successful events for statistical analysis, by mere analog simulation, in some cases requires an unreasonably long time, even on a supercomputer. Biasing schemes or importance sampling processes, such as will be used here, have been used in Monte Carlo calculations to reduce the sample size by excluding those samples which do not contribute appreciably to the quantity of interest. For schemes depending on parameter choices, as the proposed one does, the variance is a very sensitive function of these choices. Improper choices will lead to infinite variances, or to effective biases in the results, even though the theoretical bias may be zero.

As Douglass[5] observed, AI researchers are finding, for many problems, that they must couple symbolic computation with large-scale numerical calculation. This heuristic-algorithm coupling can be implemented several ways. One method of implementation for predominantly procedural analyses is to allow an algorithmic process the ability to execute a heuristic routine when necessary to perform a small amount of symbolic reasoning for some part of the algorithm. Another approach is useful for providing "deep knowledge" to symbolic modelling systems. Deep knowledge is fundamental knowledge of a system or process.[9] One form of deep knowledge is "casual knowledge" which uses causal links and general laws of nature to explain observations by reasoning from first principles. An example of this approach is Qualitative Physics for performing common-sense reasoning about the physical world.[10] Another representation of deep knowledge is numerical modelling for the purpose of simulating the device or system under study. This modelling, though, can be implemented with a conventional algorithmic computer programming language. Production Rule systems would then make use of these modelling algorithms for the purpose of acquiring deep knowledge of the system or process under study. Another approach to coupling, of which this research may be categorized, is useful for solving adaptive problems. Here, this involves large-scale numerical calculations which require occasional symbolic reasoning to guide the adaptive process. The algorithmic process is first executed, next the heuristic process analyzes the results generated so far and recommends a new path for the algorithm. The algorithm is then executed again. This process continues until the heuristic adaptive part of the system is satisfied with the solution reached based on some convergence or optimality criteria.

IMPLEMENTATION ON A GENERAL-PURPOSE SUPERCOMPUTER

The HAL-1987 General Purpose Production-Rule Analysis System is used for the heuristic part of the proposed methodology. It was developed as a tool for the generation of Model-Based systems on a supercomputer.[6] It is programmed in Portable Standard Lisp (PSL). Versions of the system currently operate under the Cray Operating System (COS), as well as the Cray Time-Sharing System (CTSS).

The system is built within the framework of an induction-deduction oriented antecedent-consequent logic, and is based on the Rule-Based System paradigm. The system uses backward-chaining corresponding to deductive logic, and forward-chaining corresponding to inductive logic.

Forward-chaining rules are actuated by each collection of facts that satisfy their IF-part, regardless of the applications's current goals. It is a useful technique for responding to unanticipated situations and for solving problems where the form of the solution is not well understood. While forward-chaining is data-oriented, backward-chaining is goal-oriented. Backward-chaining rules reduce current application rules into easier, simpler-to-achieve subgoals. Conceptually, the inference engine searches for the THEN-part of each rule to identify those rules that might conclude a result that could resolve a current goal. By limiting consideration of rules to only those that might achieve the current goals, backward-chaining frequently provides more focused problem-solving than does forward-chaining.

We consider the transport of neutral particles into a thick slab shield.[1]

PREPROCESSING OF INFORMATION

Before beginning the Monte Carlo analysis, the necessary input data such as the simulated slab thickness, etc. must be specified. The approach adopted was to have the HAL-1987 backward-chaining production-rule system query the user for the necessary information, and when completed; construct the job control file with this information included.

One of the advantages of production-rule systems is the ability to more easily implement the querying process that involves extensive branching based on the values of the user's responses. The first rule of this knowledge base, as shown in Fig. 1, has the ability to conclude the sole hypothesis (conclusion is reached) and specifies all the parameters whose values must be established and then written to the data portion of the job control file. It is up to the rest of the rules to determine these parameters' values. Figure 2 shows the possible parameters whose values the user could be asked to provide, along with their associated prompts and either acceptable values or keyword representing their acceptable range of values. As an example, "posinteger" means the only allowable value for this parameter is a positive integer. If the user attempts to enter a value different from this it will be rejected and the user will be reprompted. The third prompt shown in the example querying session of Fig. 3 demonstrates the ability of the user to either accept a default value, or reject it and enter a different value when prompted. Another useful feature is that, as shown in Fig. 3, when the querying process is complete the user can view the parameters to verify the correctness of their values.

The power of this approach is primarily enabled by the traits of the already-constructed inference engine rather than due specifically to the rules constructed. the implication is that the construction of a powerful querying system whose goals must change in response to the particular answers the user provides, can be made easier when constructed as a production-rule analysis system.

IN-CORE PROCESSING

The coupled system to be developed is to iterate between the procedural Monte Carlo simulation, executed by the job-control/data file name "montjcl", and the heuristic production-rule evaluation process, executed by the file name "updlsp", as illustrated in the lower right portion of Fig. 4. The number on the arrows denotes the order of the execution step. Steps 4 through 8 proceed in an iterative fashion until a solution is found, then exit occurs through step 9. This iterative process is controlled by using a job control file named "batchjcl".


```
((rule setup1
      (if   (* (not (filep "montjcl")))
            (xnmfp   is known)
            (sgssgt  is known)
            (alpha   is known)
            (nalpha  is known)
            (maxcoll is known)
            (wtlow   is known)
            (maxhist is known))
      (then (iteration# is 1)
            (node-refinement# is 0)
            (setup is complete)
            (* (write-montjcl))))

 (rule setup2
      (if   (alpha-default is yes))
      (then (alpha is (.20 .40 .60 .80 1.00))))

 (rule setup3
      (if   (alpha is known))
      (then (nalpha is (* (length (get-value 'alpha))))))

 (rule setup4
      (if   (maxcoll-default is yes))
      (then (maxcoll is 60)))

 (rule setup5
      (if   (wtlow-default is yes))
      (then (wtlow is 1.0e-25)))

 (rule setup6
      (if   (maxhist-default is yes))
      (then (maxhist is 100)))

 (rule setup7
      (if   (* (filep "montjcl")))
      (then (montjcl is already-existing)))  )


((setup is complete)
 (montjcl is already-existing)  )
```

Fig. 1 Rules of Knowledge Base which queries the user to formulate the problem

```
((xnmfp
  "Please enter the 'mean-free-path' thickness of the slab"
  positive)

 (sgssgt
  "Please enter the ratio of sigma-scatter/sigma total "
  posfraction)

 (alpha-default
  "Would you like to use the default values for alpha, (.20 .40 .60 .80 1.00)?"
  (yes no))

 (alpha
  "Please enter your selection of 5 alphas as a parenthesized list"
  alpha-check)

 (maxcoll-default
  "Would you like to use the default value of MAXCOLL, 60? "
  (yes no))

 (maxcoll
  "Please enter the maximum number of collisions to allow, MAXCOLL "
  posinteger)

 (wtlow-default
  "Would you like to use the default value of the lower weight limit, 1.0e-25?"
  (yes no))

 (wtlow
  "Please enter the lower weight limit at which to ignore tracks, WTLOW "
  posfraction)

 (maxhist-default
  "Is 100 histories O.K. for the first iteration? "
  (yes no))

 (maxhist
  "Please enter the number of histories for the first run "
  posinteger))
```

Fig. 2 Input parameters, their prompts, and their acceptable values or ranges


```
********* HAL is now Backward-Chaining ****************
the knowledge base used to set up the first monte carlo run

Please enter the 'mean-free-path' thickness of the slab
Please type in a positive number: >30.

Please enter the ratio of sigma-scatter/sigma total
Please type in a number between 0 and 1: >.1

Would you like to use the default values for alpha, (.20 .40 .60 .80 1.00)?
     1. YES
     2. NO
Please enter the number of your response: >1

Would you like to use the default value of MAXCOLL, 60?
     1. YES
     2. NO
Please enter the number of your response: >1

Would you like to use the default value of the lower weight limit, 1.0e-25?
     1. YES
     2. NO
Please enter the number of your response: >1

Is 100 histories O.K. for the first iteration?
     1. YES
     2. NO
Please enter the number of your response: >1

I have been able to deduce that
SETUP IS COMPLETE

********* HAL has completed Backward-Chaining *********

Would you like to see the accumulated facts?
     1. YES
     2: NO
Please enter the number of your response: >1

Here are the accumulated facts:
XNMFP IS 30.00000000
SGSSGT IS 1.00000000e-01
ALPHA-DEFAULT IS YES
ALPHA IS (.20000000 .40000000 .60000000 .80000000 1.00000000)
NALPHA IS 5
MAXCOLL-DEFAULT IS YES
MAXCOLL IS 60
WTLOW-DEFAULT IS YES
WTLOW IS 1.00000000e-25
MAXHIST-DEFAULT IS YES
MAXHIST IS 100
ITERATION# IS 1
NODE-REFINEMENT# IS 0
SETUP IS COMPLETE
```

Fig. 3 A user's interactive session with the querying Production Rule system

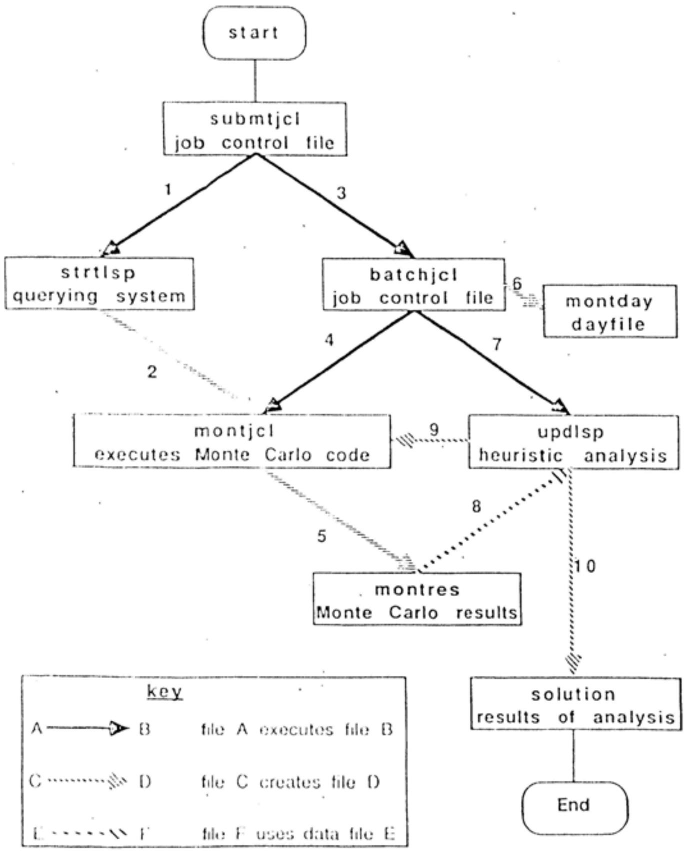


Fig 4 Control Structure of the coupled system

That file accomplishes this task as follows: The Monte Carlo simulation and the heuristic evaluation are able to communicate by the use of two data files. The Monte Carlo code uses the job control file "montjcl" both to get itself executed and as a source of data for the simulation. The results of the simulation are recorded in the file "montres". The production-rule system reads through the newly created montres file to evaluate the latest simulation, and sets up a new simulation by creating a new montcl file with the proper simulation parameters values. It is up to the job control file batchjcl to orchestrate this iteration.

One interesting note is as follows: This file structure has been organized to allow the alternate execution of two specific programs. Batchjcl always executes the production-rule system with the file updlsp and then executes the job control file montjcl. However, under this setup, if the production-rule system found it necessary to have some other program or programs executed, this could be accomplished by having it still create a file by the name of montjcl, but instead of putting in it the instructions to execute the same Monte Carlo simulation, put in the instructions to execute whatever program(s) is or are desired. The file montjcl will be executed as always, and all that is required is that the production-rule system knows what files those programs created and what to do with them. This feature will enable future expansion and modifications to be made to the present system, as well as make feasible the coupling of more complex systems.

As presently configured, batchjcl repeatedly executes the two files montjcl and updlsp. It repeatedly deletes the job control/data file montjcl and the data file montres after each use to make room for the next versions of these files to be created by the production-rule system and the Monte Carlo sumulation, respectively. This iteration stops when batchjcl detects that the production-rule system hasn't created the file montjcl. This indicates that the heuristic process has found a solution. When it finds a solution, the production-rule system no longer creates a montjcl file and instead writes the solution to the file "solution". A solution file is shown in Fig. 5. In order to keep a record of the iteration process being performed, batchjcl appends each montres file to a file called "montday". Thus, at the end of an analysis session the user has not only the solution, but also the results of each of the Monte Carlo simulations carried out. This

```
The transmission probability has been found to converge to 1.59076700e-05
for a 20.00000000 mean-free-path thickness slab
with a scattering probability of .70000000
This occurred for an alpha value of .52500000 and 25000 histories.
The final value was arrived at after 8 iterations and 3 mesh refinements.

Date of run: 870216
Job started (hhmm): 2046
Job finished (hhmm): 2057
Computer time used(in minutes): .798
```

Fig. 5 Results of an analysis in the file "solution"


```
(rule update2
  (if   (* (lessp (get-value 'iteration#) 20))
        (* (lessp (get-value 'maxhist) 100000)))
  (then (endpoint is not-reached)))

(rule update3
  (if   (* (geq (get-value 'iteration#) 20)))
  (then (endpoint is reached)))

(rule update4
  (if   (* (geq (get-value 'maxhist) 100000)))
  (then (endpoint is reached)))

(rule update5
  (if   (pattern is (4 1)))
  (then (update-type is 1)))

(rule update6
  (if   (pattern is (1 4)))
  (then (update-type is 1)))

(rule update7
  (if   (pattern is (1 3 1)))
  (then (update-type is 1)))

(rule update8
  (if   (pattern is (3 2)))
  (then (update-type is 2)))

(rule update9
  (if   (pattern is (2 3)))
  (then (update-type is 2)))

(rule update10
  (if   (pattern is (e 5)))
  (then (update-type is 3)))

(rule update11
  (if   (pattern is (5 e)))
  (then (update-type is 3)))

(rule update12
  (if   (pattern is (e 4)))
  (then (update-type is 3)))

(rule update13
  (if   (pattern is (4 e)))
  (then (update-type is 3)))

(rule update14
  (if   (pattern is (e 3)))
  (then (update-type is 3)))

(rule update15
  (if   (pattern is (3 e)))
  (then (update-type is 3)))

(rule update16
  (if   (pattern is other))
  (then (update-type is 4)))

(rule update17
  (if   (pattern is (c 5 c)))
  (then (update-type is 5)))

(rule update18
  (if   (update-type is 1)
        (endpoint is not-reached)
        (* (lessp (get-value 'node-refinement#) 5)))
  (then (alpha is (* (refine-mesh1)))
        (iteration# is (* (add1 (get-value 'iteration#))))
        (node-refinement# is
           (* (add1 (get-value 'node-refinement#))))
        (maxhist is (* (increment-maxhist)))
        (iteration-type is refine-mesh1)))

(rule update19
  (if   (update-type is 1)
        (endpoint is not-reached)
        (* (geq (get-value 'node-refinement#) 5)))
  (then (iteration# is (* (add1 (get-value 'iteration#))))
        (maxhist is (* (increment-maxhist)))
        (iteration-type is run-longer)))

(rule update20
  (if   (update-type is 1)
        (endpoint is reached))
  (then (* (punt))
        (solution is punted)))

(rule update21
  (if   (update-type is 2)
        (endpoint is not-reached)
        (* (lessp (get-value 'node-refinement#) 5)))
  (then (alpha is (* (refine-mesh2)))
        (iteration# is (* (add1 (get-value 'iteration#))))
        (node-refinement# is
           (* (add1 (get-value 'node-refinement#))))
        (maxhist is (* (increment-maxhist)))
        (iteration-type is refine-mesh2)))

(rule update22
  (if   (update-type is 2)
        (endpoint is not-reached)
        (* (geq (get-value 'node-refinement#) 5)))
  (then (iteration# is (* (add1 (get-value 'iteration#))))
        (maxhist is (* (increment-maxhist)))
        (iteration-type is run-longer)))

(rule update23
  (if   (update-type is 2)
        (endpoint is reached))
  (then (* (punt))
        (solution is punted)))

(rule update24
  (if   (update-type is 3)
        (endpoint is not-reached))
  (then (alpha is (* (shift-mesh3)))
        (iteration# is (* (add1 (get-value 'iteration#))))
        (iteration-type is shift-mesh3)))

(rule update25
  (if   (update-type is 3)
        (endpoint is reached))
  (then (* (punt))
        (solution is punted)))

(rule update26
  (if   (update-type is 4)
        (endpoint is not-reached))
  (then (iteration# is (* (add1 (get-value 'iteration#))))
        (maxhist is (* (increment-maxhist)))
        (iteration-type is run-longer)))

(rule update27
  (if   (update-type is 4)
        (endpoint is reached))
  (then (* (punt))
        (solution is punted)))

(rule update28
  (if   (update-type is 5))
  (then (* (prepare-solution))
        (solution is reached)))

(rule update29
  (if   (iteration-type is known))
  (then (* (write-montjcl))
        (solution is delayed))) )

((solution is reached)
 (solution is punted)
 (solution is delayed))
```

Fig. 6 Rules of Variance Reduction Knowledge Base

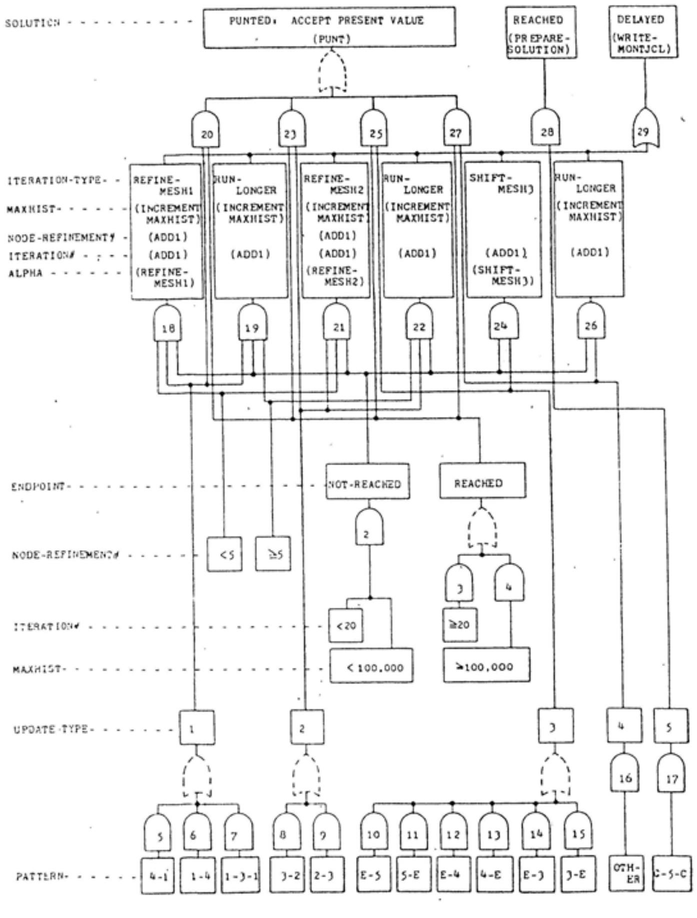


Fig. 7 Variance reduction decision process as a Goal Tree

also opens up the possiblity of future postprocessing which could graphically show the user the refinement of the solution.

The iteration process described must be initiated somehow, and it is also necessary to execute the front-end production-rule querying system before iteration begins. Therefore this process is structured as follows: A job control file by the name of "submtjcl" first executes the production-rule querying system, which will poll the user and put that data into the file montjcl. After supplying the system with the necessary information to set up the problem, intervention by the user is no longer needed. Iteration is ready to begin, and it can be carried out as a batch process. Therefore submtjcl executes the non-interactive job control file batchjcl for controlling the simulation/evaluation process that was described above. When a solution is reached, the heuristic Monte Carlo results evaluator creates a solution file instead of a montjcl file, the batchjcl job control file halts the iteration process, and control reverts back to the submtjcl job control file. As its last act, submtjcl appends to the solution file the amount of cpu time and wall-clock time required to find this solution. And all execution stops.

The Production-Rule system uses HAL-1987 in a non-interactive forward-chaining mode. The rules for this Knowledge Base are shown in Fig. 6, and also shown as a goal-tree in Fig. 7. Because it is operating in a batch mode and thus cannot rely on the user to type in the input data, the Production-Rule system has the ability to take information from a data file and perform any necessary preprocessing steps before the forward-chaining proces begins. Thus the results from the simulation are read from the Monte Carlo simulation results file, montres, and placed into the pool of known facts. The smallest number in each row of the second-moment array is found and then, for the whole array, the number of minima found in each column is also recorded.

DEMONSTRATIVE CASES AND EFFICIENCY COMPARISONS

To assess the capabilities of the proposed methodology, the solutions to problems with different slab thicknesses were computed. Three cases with different media properties were considered:

Problem A: $P_s=\sum_s/\sum_t=0.9$ Highly Scattering Medium
Problem B: $P_s=\sum_s/\sum_t=0.5$ Moderately Absorbing Medium
Problem C: $P_s=\sum_s/\sum_t=0.1$ Highly Absorbing Medium

The results from the proposed methodology are then compared to the exact results obtained from the analytical solution to the problems, as well as to the results obtained through applying the straightforward methodology without the adaptive hybrid symbolic-procedural search.

Figure 8 shows the Production-Rule System's interaction with the user for problem setup for a 30 mean free paths thick semi-infinite slab and $\sum_s/\sum_t=0.1$, corresponding to a highly absorbing medium. The Inference Engine in this situation functions in the backward-chaining mode and queries the user for the data required for the calculation, and concurrently tests their validity. This preprocessing stage acts as an intelligent manual since it would flag any inconsistencies in the input information. After supplying the necessary input data, the user has chosen to verify its correctness by viewing the "accumulated facts". Once the system has obtained the necessary input data, the rest of the process proceeds without user intervention under the control of the rules that the user has implemented for this purpose in the knowledge base. When a solution is reached, the results are placed in the file solution. This file includes the transmission probability result, the associated computed optimal variance reduction parameter, as well as the computed time statistics. Table 1 summarizes the results obtained by the proposed approach, for problems A, B, and C for a slab thickness of 20 mean free paths, and compares them to the exact analytical results. It also shows the results obtained by application of the straightforward approach without the use of the adaptive methodology for the five guesses of α: 0.10, 0.35, 0.50, 0.75, and 1.00. The latter value corresponds to the analog process. The cases are run for 10,000 and 100,000 particle histories. The result is shown as:

$$\mu \pm \sigma$$

where: μ is the mean
σ is the standard deviation of the mean.

Notice how in situations where the guess for the variance reduction parameter is far from the optimal value, the straightforward application of biasing fails in even obtaining the right order of magnitude of the result. For instance, consider the case:


```
********* HAL is now Backward-Chaining ****************
the knowledge base used to set up the first monte carlo run

Please enter the 'mean-free-path' thickness of the slab
Please type in a positive number: >30.

Please enter the ratio of sigma-scatter/sigma total
Please type in a number between 0 and 1: >.1

Would you like to use the default values for alpha, (.20 .40 .60 .80 1.00)?
     1. YES
     2. NO
Please enter the number of your response: >1

Would you like to use the default value of MAXCOLL, 60?
     1. YES
     2. NO
Please enter the number of your response: >1

Would you like to use the default value of the lower weight limit, 1.0e-25?
     1. YES
     2. NO
Please enter the number of your response: >1

Is 100 histories O.K. for the first iteration?
     1. YES
     2. NO
Please enter the number of your response: >1

I have been able to deduce that
SETUP IS COMPLETE

********* HAL has completed Backward-Chaining *********

Would you like to see the accumulated facts?
     1. YES
     2. NO
Please enter the number of your response: >1

Here are the accumulated facts:
XNMFP IS 30.00000000
SGSSGT IS 1.00000000e-01
ALPHA-DEFAULT IS YES
ALPHA IS (.20000000 .40000000 .60000000 .80000000 1.00000000)
NALPHA IS 5
MAXCOLL-DEFAULT IS YES
MAXCOLL IS 60
WTLOW-DEFAULT IS YES
WTLOW IS 1.00000000e-25
MAXHIST-DEFAULT IS YES
MAXHIST IS 100
ITERATION# IS 1
NODE-REFINEMENT# IS 0
SETUP IS COMPLETE
```

Fig. 8 Interaction with user for case of 30 mean free paths with $\Sigma_s/\Sigma_t=0.1$ for highly absorbing medium

Table 1 Mean values, standard deviations, and cpu times for the comparative cases for 20 mean free paths

The results for the proposed adaptive approach are:

A: $\mu = 1.33568\times10^{-3} \pm 0.02125\times10^{-3}$, $T = 1.967$ min, $\mu_{exact} = 1.30823\times10^{-3}$

B: $\mu = 6.978529\times10^{-7} \pm 0.4245\times10^{-7}$, $T = 0.327$ min, $\mu_{exact} = 7.001165\times10^{-7}$

C: $\mu = 5.786251\times10^{-9} \pm 0.0648\times10^{-9}$, $T = 0.803$ min, $\mu_{exact} = 5.748289\times10^{-9}$

$\mu \equiv$ mean value

$\sigma \equiv$ standard deviation

$T \equiv$ CPU time (minutes) on Cray X-MP

	Non-adaptive Approach data					
Problem	Number of histories	Guess for variance reduction parameter α				
		0.10	0.35	0.50	0.75	1.00
		$\mu \pm \sigma$ T	$\mu \pm \sigma$ T	$\mu \pm \sigma$ T	$\mu \pm \sigma$ T	$\mu \pm \sigma$ T
A	10,000	0.069142e-3 ±0.0370e-3 0.009	0.850471e-3 ±0.1264e-3 0.023	1.209853e-3 ±0.0759e-3 0.032	1.359153e-3 ±0.0480e-3 0.036	1.405785e-3 ±0.1271e-3 0.025
	100,000	0.206383e-3 ±0.1025e-3 0.078	1.160779e-3 ±0.1076e-3 0.223	1.243661e-3 ±0.0294e-3 0.303	1.302449e-3 ±0.0148e-3 0.351	1.336936e-3 ±0.0381e-3 0.244
B	10,000	4.029975e-7 ±0.5152e-7 0.009	7.373355e-7 ±0.3616e-7 0.023	7.349772e-7 ±0.4945e-7 0.032	9.180513e-7 ±2.1193e-7 0.036	6.046890e-7 ±4.0562e-7 0.025
	100,000	5.846845e-7 ±5.8646e-7 0.078	7.1066707e-7 ±0.1250e-7 0.223	7.340175e-7 ±0.1561e-7 0.303	6.856662e-7 ±0.5911e-7 0.350	4.278244e-7 ±1.1288e-6 0.243
C	10,000	6.026776e-9 ±0.2107e-9 0.009	3.995931e-9 ±0.6797e-9 0.021	3.715632e-9 ±1.4734e-9 0.024	0.347734e-9 ±0.2422e-9 0.022	0.012011e-9 ±0.0101e-9 0.017
	100,000	5.811901e-9 ±0.0651e-9 0.078	5.759100e-9 ±0.3363e-9 0.205	5.897306e-9 ±1.2844e-9 0.235	0.373400e-9 ±0.1793e-9 0.215	0.011640e-9 ±0.0100e-5 0.163

Problem C, 10,000 histories, $\alpha=0.75$, 20 mfps leading to a result:

$$(\mu\pm\sigma)_{10,000} = (0.3477\times10^{-9}\pm0.2422\times10^{-9})$$

which is an order of magnitude lower than the exact result:

$$\mu_{exact}=5.7483\times10^{-9}.$$

Increasing the number of histories to 100,000 histories leads to:

$$(\mu\pm\sigma)_{100,000}=(0.3734\times10^{-9}\pm0.1793\times10^{-9})$$

which is still lower than the exact value by an order of magnitude. The variance estimates are undependable since they are underestimates too. Application of the proposed methodology leads to the satisfactory result:

$$(\mu\pm\sigma)=(5.7863\times10^{-9}\pm0.648\times10^{-9})$$

which is also associated with a smaller standard deviation of the mean. This is evidently obtained at the expense of more computations where T=0.803 minutes compared with 0.215 minutes for the non-adaptive approach.

A valid performance comparison between the adaptive control and the conventional method must thus consider both the improved accuracy and the time expended in reaching a given result. To carry out such a comparison, it is clear one must carry the comparison relative to the exact result and not relative to the estimated variances. For this purpose we define a figure of merit as the ratio of the efficiency of a methodology 2 to the efficiency of a methodology 1 is given by:

$$F_{21} = \frac{\eta_2}{\eta_1} = \left(\frac{\Lambda_1}{\Lambda_2}\right)\left(\frac{T_1}{T_2}\right) \qquad (1)$$

where
- Λ_1/Λ_2 is the error ratio
- T_1/T_2 is the labor ratio
- T_i is the time consumed by method i
- Λ_i is the error estimate for method i
- η_i is the efficiency of method i:

$$\eta_i \propto \frac{1}{\Lambda_i T_i}$$

The error estimate Λ_i can be defined as the relative error compared with the exact result:

$$\Lambda_i^1 = \frac{|V_i - V_E|}{V_E} \tag{2}$$

where V_E is the exact mean value and V_i is the estimated mean value by method i. In this case Eq. 1 becomes:

$$F_{21}^1 = \frac{|V_i - V_E|}{|V_2 - V_E|} \left(\frac{T_1}{T_2}\right) \tag{3}$$

where 2 refers to the proposed methodology and 1 refers to the non-adaptive method. Table 2 shows the value of this efficiency ratio computed from the results from Table 1 for 20 mean free paths. It can be observed that if a user would have initially guessed the optimal value of his variance reduction parameter (e.g. 0.35 for Problem B), then the efficiency ratio has a low value (3.2 for 100,000 histories). The advantage of the proposed methodology is apparent if the user would have guessed values for α_i that are far from the optimum. For the guesses 0.1, 0.5, 0.75, and 1.00 the efficiency ratios are, 12, 14, 7, and 89 respectively for the case of 100,000 histories.

It can be noticed in Table 2 that the improvement in the efficiency ratio by application of the adaptive methodology appears only for moderately and highly absorbing media. For highly scattering media, and thin medium thicknesses, the straightforward approach achieves satisfactory results. To further investigate the range of applicability of the adaptive methodology we consider a harder to solve deep-penetration problem of 40 mean free paths rather than only 20. The results for this case are shown in Table 3. It can be noticed that even with a large number of histories (100,000) the results of the straightforward method are several orders of magnitude lower than the exact result. For instance:
Problem C, 100,000 histories, α=0.75, 40 mfps
has a result

$$(\mu \pm \sigma)_{100,000} = (4.7522\times10^{-27} \pm 2.9972\times10^{-27})$$

which is here ten orders of magnitude lower than the exact mean value:

$$\mu_{exact} = 3.3066\times10^{-17}.$$

Table 2 F^1_{21} Efficiency Ratios comparing the adaptive approach to the non-adaptive approach for a 20 mean free paths slab for three problems where $P_s=\Sigma_s/\Sigma_t$

A: $P_s = 0.9$ Highly Scattering Medium, $\alpha_{opt} = 0.75$
B: $P_s = 0.5$ Moderately Absorbing Medium $\alpha_{opt} = 0.35$
C: $P_s = 0.1$ Highly Absorbing Medium $\alpha_{opt} = 0.10$

Problem	Non adaptive Approach data					
	Number of histories	Guess for variance reduction parameter α				
		0.10	0.35	0.50	0.75	1.00
A	10,000	0.2	0.2	0.1	0.0	0.0
	100,000	1.6	0.6	0.4	0.0	0.1
B	10,000	3.6	1.2	1.5	10.6	3.2
	100,000	12.2	3.2	13.9	6.8	89.4
C	10,000	0.1	1.2	1.6	3.9	3.2
	100,000	0.2	0.1	1.2	37.9	30.7

Table 3 Mean values, standard deviations, and cpu times for the comparative cases for 40 mean free paths

The results for the proposed adaptive approach are:

A: $\mu = 1.675339\times10^{-3} \pm 0.1465\times10^{-6}$, $T = 1.007$ min, $\mu_{exact} = 2.344007\times10^{-6}$

B: $\mu = 2.933906\times10^{-13} \pm 0.6010\times10^{-13}$, $T = 0.291$ min, $\mu_{exact} = 5.050341\times10^{-13}$

C. $\mu = 2.219680\times10^{-17} \pm 0.3964\times10^{-17}$, $T = 0.247$ min, $\mu_{exact} = 3.306602\times10^{-17}$

$\mu \equiv$ mean value

$\sigma \equiv$ standard deviation

$T \equiv$ CPU time (minutes) on Cray X-MP

Problem	Non-adaptive Approach data					
	Number of histories	Guess for variance reduction parameter α				
		0.10	0.35	0.50	0.75	1.00
		$\mu \pm \sigma$ T	$\mu \pm \sigma$ T	$\mu \pm \sigma$ T	$\mu \pm \sigma$ T	$\mu \pm \sigma$ T
A	100,000	0.004533e-6 ±0.0031e-6 0.139	0.963956e-6 ±0.1573e-6 0.426	1.518471e-6 ±0.1278e-6 0.528	1.549895e-6 ±0.0678e-6 0.441	1.927235e-6 ±0.3815e-6 0.253
B	100,000	1.832358e-13 ±0.2142e-13 0.139	5.374517e-13 ±0.3272e-13 0.435	5.124573e-13 ±0.6728e-13 0.540	4.182849e-13 ±2.1937e-13 0.447	2.623639e-17 ±1.8809e-17 0.255
C	100,000	3.354181e-17 ±0.0617e-17 0.135	5.530385e-17 ±3.5254e-17 0.270	0.229973e-17 ±0.2133e-17 0.266	4.752200e-27 ±2.9972e-27 0.221	0.0 -- 0.163

Table 4 F^1_{21} Efficiency Ratios comparing the adaptive approach to the non-adaptive approach for a 40 mean free paths slab for three problems where $P_s = \Sigma_s/\Sigma_t$

A: $P_s = 0.9$ Highly Scattering Medium, $\alpha_{opt} = 0.73$

B: $P_s = 0.5$ Moderately Absorbing Medium $\alpha_{opt} = 0.36$

C. $P_s = 0.1$ Highly Absorbing Medium $\alpha_{opt} = 0.07$

Problem	Non-adaptive Approach data					
	Number of histories	Guess for variance reduction parameter α				
		0.10	0.35	0.50	0.75	1.00
A	100,000	0.5	0.9	0.6	0.5	0.2
B	100,000	0.7	0.2	0.1	0.6	2.1
C	100,000	0.0	2.2	3.1	2.7	∞

The proposed methodology's estimate is:

$$(\mu\pm\sigma)_{100,000} = (2.2197\times10^{-17}\pm0.3964\times10^{-17})$$

which captures with 0.247 minutes the right order of magnitude of the result. Table 4 displays the values of the efficiency ratios of Eq. 3 and shows that its value is zero if the user happens to guess a value of his parameter very close to the optimal value (e.g. $\alpha=0.1$ which is close to $\alpha_{opt}=0.07$). However it also shows that if the user would have used the analog value ($\alpha=1$) then the efficiency ratio is infinite implying that a solution to the problem cannot be obtained by the choice of $\alpha=1$, at least within the 100,000 histories run.

CONCLUSION

A Hybrid Symbolic-Procedural Programming Adaptive methodology is developed for the solution of stochastic simulations problems on a general-purpose supercomputer for scientific computations. The usefulness of the proposed methodology becomes apparent when the adptive methodology requires a construct that is heuristic in nature, and lends itself to symbolic programming, whereas the simulation is stochastic in nature, and lends itself to procedural programming. Monte Carlo particle transport simulations are here considered where the simulation is conducted using procedural programming in Fortran, and the variance reduction process uses symbolic programming in Portable Standard Lisp (PSL).

The Monte Carlo methodology uses importance sampling with parametric dependence, and obtains over each single stage of a multistage process the overall functional dependence of the variance upon a variance reduction parameter α over a broad range of its values. Results corresponding to minimum variance α_{opt} are adopted and others are rejected. The use of the exponential transformation with the parameter α here involves a scaling of the particle tracks in a new application of Morton's method of similar trajectories. The method generalizes Spanier's multistage importance sampling method through preservation of the statistical correlations between the particle tracks according to a theory due to Frolov and Chentsov. The methodology avoids the effective biases and infinite variances observed in this type of calculation whenever the choices of the variance reduction parameters are far from their optimal values.

Because the optimization process for the variance reduction parameters lends itself to heuristic analysis, a general-purpose inference engine, HAL-1987, was developed for the analysis of the heuristic aspects of the methodology. It uses the Rule-Based paradigm with an antecedent-consequent logic and is capable of forward- and backward-chaining in an inductive-deductive logic environment. The solution is implemented through the construction of a Production-Rule System for Analysis which carries out the preprocessing of the input data to the problem, then conducts the adaptive control of the optimization of the variance reduction parameters in the simulation using a Knowledge Base encompassing the expertise of the user. The need for multiple job submissions is eliminated, and a learning process in the search for the optimum is generated while the job is executing.

The performance of the proposed methodology is compared with that of the non-adaptive particle transport simulation approach using importance sampling, and to derived exact analytical solutions for the case of a semi-infinite slab for highly and moderately absorbing and highly scattering media, through the efficiency ratio defined as the product of the labor and error ratios. If a user would have initially guessed the optimal values of the variance reduction parameters, no substantial increase in efficiency is observed. However, for the common case where the user may have initially guessed values far from the optimum, efficiency increases are observed. For a 20 mean free paths moderately absorbing medium, and 100,000 histories with an exact analytical optimum value of the variance reduction parameter at $\alpha_{opt}=0.35$, if the user would have guessed the optimal value, and efficiency ratio of 3 is obtained. However, if he would have guessed 0.1, 0.5, 0.75, or 1.0, then application of the proposed methodology would lead to efficiency ratios of 12, 14, 7 and 89 respectively, relative to the non-adaptive approach.

ACKNOWLEDGEMENTS

The financial support from the National Science Foundation the Illinois Department of Nuclear Safety (IDNS) and the National Center for Supercomputing Applications (NCSA) is acknowledged. The input from Prof. L. Smarr from the NCSA, Mike Parker from IDNS, George Miley from the Department of Nuclear Engineering and Prof. M. Harandi from the Computer Science Dept., University of Illinois is appreciated. Thanks are due to Kathy Dysart for the manuscript preparation.

REFERENCES

1. Ragheb, M.M.H., (1981) "Monte Carlo Parametric Importance Sampling with Particle Tracks Scaling," Atomkernenergie/Kerntechnik, 39, No. 3, 198-206.

2. Shortliffe, E., (1984) "Computer-Based Medical Consultants: MYCIN," Elsevier, New York.

3. Duda, R., P. Hart, P. Barrett, et al., (1978) "Development of the PROSPECTOR System for Mineral Exploration," SRI Report Projects 5822 and 6415, Standord Research Institute, Palo Alto, CA.

4. Feigenbaum, E.A., and P. McCorduck, (1984) "The Fifth Generation," New American Library, New York.

5. Douglass, R.J., (1985) "Artifical Intelligence Applications on Supercomputers". 1985 Cray Science and Engineering Symposium, Minneapolis, MN.

6. Ragheb, M., and D. Gvillo, (1986) "Symbolic Simulation of Engineering Systems on a Supercomputer," Proc. Conf. on "Applications of Artificial Intelligence III," SPIE Proceedings, Vol. 635, Florida.

7. Ragheb, M.M.H., (1984) "Lecture Notes on Monte Carlo Simulation," Nuclear Engineering Program, PSL-89, University of Illinois, Urbana, IL.

8. Halton, J.H., (1970) "A Retrospective and Prospective Survey of the Monte Carlo Method," SIAM Review, Vol. 12, No. 1, January.

9. Buchanan, B.G., and E.H. Shortliffe, (1984) "Rule-Based Expert Systems," Addison-Wesley, Reading, Massachusetts.

10. Forbus, K.D., (1984) "Qualitative Process Theory," TR-789, MIT Artificial Intelligence Lab., Cambridge, Mass.