Monte Carlo Synthetic Acceleration Methods for Sparse Linear Systems

Massimiliano Lupo Pasini (Emory University)

In collaboration with:
Michele Benzi (Emory University)
Thomas Evans (Oak Ridge National Laboratory)
Steven Hamilton (Oak Ridge National Laboratory)
Stuart Slattery (Oak Ridge National Laboratory)

SIAM Conference on Applied Linear Algebra 2015

October 30, 2015

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research program.

Outline

- Motivations

Outline

- Motivations
- Standard Monte Carlo vs. Monte Carlo Synthetic Acceleration (MCSA)

Outline

- Motivations
- Standard Monte Carlo vs. Monte Carlo Synthetic Acceleration (MCSA)
- Choice of the preconditioner

Outline

- Motivations
- Standard Monte Carlo vs. Monte Carlo Synthetic Acceleration (MCSA)
- Choice of the preconditioner
- Convergence conditions

Outline

- Motivations
- Standard Monte Carlo vs. Monte Carlo Synthetic Acceleration (MCSA)
- Choice of the preconditioner
- Convergence conditions
- Numerical results

Motivations

Goal:

- development of reliable algorithms to solve highly dimensional (towards exascale) sparse linear systems in parallel

Issues:

- occurrence of **faults** (abnormal operating conditions of the computing system which cause a wrong answer)
  - hard faults
  - soft faults

[M. Hoemmen and M.A. Heroux, 2011; G. Bronevetsky and B.R. de Supinski, 2008; P. Prata and J.B. Silva, 1999]

Motivations

Resilience: ability to compute a correct output in the presence of faults

Different approaches to tackle the problem:

- adaptation of Krylov subspace methods (CG, GMRES, Bi-CGStab, ...) to fault tolerance by recover-restart strategies
- abandonment of deterministic paradigm in favor of stochastic approaches

[E. Agullo, L. Giraud, A. Guernouche, J. Roman and M. Zounon, 2013]

Monte Carlo Linear Solvers

Mathematical setting

Let us consider a sparse linear system

$$A\mathbf{x} = \mathbf{b}, \tag{1}$$

where $A \in \mathbb{R}^{n \times n}$, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^n$.
With left preconditioning (1) becomes

$$P^{-1}A\mathbf{x} = P^{-1}\mathbf{b}, \quad P \in \mathbb{R}^{n \times n}. \tag{2}$$

(2) can be reinterpreted as a fixed point scheme

$$\mathbf{x} = H\mathbf{x} + \mathbf{f}, \quad H = I - P^{-1}A, \ \mathbf{f} = P^{-1}\mathbf{b}. \tag{3}$$

Assuming $\rho(H) < 1$, (3) generates a sequence of approximate solutions $\{\mathbf{x}_k\}_{k=0}^{\infty}$ which converges to the exact solution of (1).

Monte Carlo Linear Solvers

Mathematical setting

The solution to (3) can be written in terms of a power series in H (Neumann series):

$$\mathbf{x} = \sum_{\ell=0}^{\infty} H^{\ell}\mathbf{f}.$$

By restricting the attention to a single component of $\mathbf{x}$ we have

$$x_i = f_i + \sum_{\ell=1}^{\infty} \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} \cdots \sum_{k_\ell=1}^{n} H_{i,k_1} H_{k_1,k_2} \ldots H_{k_{\ell-1},k_\ell} f_{k_\ell}. \qquad (4)$$

(4) can be reinterpreted as the sampling of an estimator defined on a random walk.

Monte Carlo Linear Solvers

Mathematical setting: Forward (Direct) Method

Goal: evaluate a functional such as

$$J(\mathbf{x}) = \langle \mathbf{h}, \mathbf{x} \rangle = \sum_{i=1}^{n} h_i x_i, \quad \mathbf{h} \in \mathbb{R}^n.$$

Approach: define random walks to evaluate J.
Consider a random walk whose state space S is the set of indices of the forcing term $\mathbf{f}$:

$$S = \{1, 2, \ldots, n\} \subset \mathbb{N}.$$

[N. Metropolis and S. Ulam, 1949; G. E. Forsythe and A. Leibler, 1950; W. R. Wasow, 1952; J. H. Halton, 1962]

Monte Carlo Linear Solvers

Mathematical setting: Forward (Direct) Method

Initial probability: pick k_0 as initial step of the random walk:

$$\tilde{p}(k_0 = i) = \tilde{p}_{k_0} = \frac{|h_i|}{\sum_{i=1}^{n} |h_i|}.$$

Possible choices for the transition probability:
$p(:, i) : S \to [0, 1] \quad \forall i \in S$

$$p(k_i = j | k_{i-1} = i) = P_{i,j} = \frac{|H_{i,j}|}{\sum_{k=1}^{n} |H_{i,k}|} \qquad \textit{weighted}$$

$$P_{i,j} = \begin{cases} 0 & \text{if} \quad H_{i,j} = 0 \\ \frac{1}{\#(\text{non-zeros in the row})} & \text{if} \quad H_{i,j} \neq 0 \end{cases} \qquad \textit{uniform}$$

Monte Carlo Linear Solvers

Mathematical setting: Forward (Direct) Method

A related sequence of weights is defined by

$$w_{i,j} = \frac{H_{i,j}}{P_{i,j}}$$

in order to build an auxiliary sequence

$$W_0 = \frac{h_{k_0}}{\tilde{p}_{k_0}}, \quad W_i = W_{i-1} w_{k_{i-1},k_i} \quad k_{i-1}, k_i \in S.$$

This allows us to define a random variable $X(\cdot) : \Pi \to \mathbb{R}$.
Π= set of realizations of a random walk γ defined on S:

$$X(\nu) = \sum_{\ell=0}^{\infty} W_\ell f_{k_\ell}, \quad \nu = \text{permutation of } \gamma$$

Monte Carlo Linear Solvers

Mathematical setting: Forward (Direct) Method

Define the expected value of $X(\cdot)$:

$$E[X] = \sum_{\nu} P_\nu X(\nu) \quad (P_\nu = \text{probability of the permutation } \nu).$$

It can be proved that

$$E[W_\ell f_{k_\ell}] = \left\langle \mathbf{h}, H^\ell \mathbf{f} \right\rangle$$

and consequently

$$E\left[\sum_{\ell=0}^{\infty} W_\ell f_{k_\ell} \right] = \langle \mathbf{h}, \mathbf{x} \rangle .$$

Monte Carlo Linear Solvers

Mathematical setting: Forward (Direct) Method

If $\mathbf{h}$ is a vector of the standard basis: $\mathbf{h} = \mathbf{e}_j$

$$\tilde{p}(K_0 = i) = \delta_{i,j}, \quad i, j = 1, \ldots, n$$

and the expected value assumes the following form:

$$E\left[\sum_{\ell=0}^{\infty} W_\ell f_{k_\ell}\right] = x_i = f_i + \sum_{\ell=1}^{\infty} \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} \cdots \sum_{k_\ell=1}^{n} P_{i,k_1} P_{k_1,k_2} \cdots P_{k_{\ell-1},k_\ell} w_{i,k_1} w_{k_1,k_2} \cdots w_{k_{\ell-1},k_\ell} f_{k_\ell}. \tag{5}$$

Drawback: the estimator is defined entry-wise.

Monte Carlo Linear Solvers

Mathematical setting: Adjoint Method

Consider the adjoint linear system

$$(P^{-1}A)^T\mathbf{y} = \mathbf{d}.$$

With a fixed point approach, we get

$$\mathbf{y} = H^T\mathbf{y} + \mathbf{d}.$$

By picking $\mathbf{d} = \mathbf{e}_j$ we have

$$J^*(\mathbf{y}) := \langle \mathbf{y}, \mathbf{f} \rangle = \langle \mathbf{x}, \mathbf{e}_j \rangle = x_j, \quad j = 1, \ldots, n.$$

Monte Carlo Linear Solvers

Mathematical setting: Adjoint Method

The initial probability and transition matrix may be defined as

$$\tilde{p}_{k0} = p(k_0 = i) = \frac{|f_i|}{\sum_{i=1}^{n} |f_i|}, \quad P_{i,j} = \frac{|H^T_{i,j}|}{\sum_{k=1}^{n} |H^T_{i,k}|} = \frac{|H_{j,i}|}{\sum_{k=1}^{n} |H_{k,i}|}.$$

Reintroduce weights as before:

$$w_{i,j} = \frac{H_{j,i}}{P_{i,j}} \Rightarrow W_0 = sign(f_i)\|\mathbf{f}\|_1, \quad W_j = W_{j-1} w_{k_{i-1},k_i}, \quad k_{i-1}, k_i \in S.$$

Expected value of the estimator for the entire solution vector $\mathbf{x}$:

$$E\left[\sum_{\ell=0}^{\infty} W_\ell d_{k_\ell}\right] = \sum_{\ell=0}^{\infty} \sum_{k_0=1}^{n} \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} \cdots \sum_{k_\ell=1}^{n} f_{k_0} P_{k_0,k_1} P_{k_1,k_2} \cdots P_{k_{\ell-1},k_\ell} w_{k_0,k_1} \cdots w_{k_{\ell-1},k_\ell} d_{k_\ell}. \quad (6)$$

Monte Carlo Linear Solvers

Statistical constraints for convergence

Expected value and variance of the estimator must be finite to apply the Central Limit Theorem.

Forward Method: $(\hat{H})_{i,j} = \frac{(H_{i,j})^2}{P_{i,j}}$

$$E\left[\sum_{\ell=0}^{\infty} W_\ell f_{k_\ell}\right] < \infty \Leftrightarrow \rho(H) < 1$$

$$Var\left[\sum_{\ell=0}^{\infty} W_\ell f_{k_\ell}\right] < \infty \Leftrightarrow \rho(\hat{H}) < 1$$

Adjoint Method: $(\hat{H})_{i,j} = \frac{(H_{j,i})^2}{P_{i,j}}$

$$E\left[\sum_{\ell=0}^{\infty} W_\ell d_{k_\ell}\right] < \infty \Leftrightarrow \rho(H) < 1$$

$$Var\left[\sum_{\ell=0}^{\infty} W_\ell d_{k_\ell}\right] < \infty \Leftrightarrow \rho(\hat{H}) < 1$$

$$\rho(\hat{H}) < 1 \Rightarrow \rho(H) < 1$$

[A. Srinivasan, Math. Comput. Simulat. 2010; J. H. Halton, 1994; H. Ji, M. Mascagni and Y. Li, 2013]

Monte Carlo Linear Solvers

First approaches to a parallelization

First attempts to apply Monte Carlo as a linear solver considered the fixed point formulation of the original linear system.

- \+ embarrassingly parallelizable (ideally we could employ as many processes as the amount of histories)
- – huge amount of time to compute a reasonably accurate solution

E.g. $\|r_0\| = \|\mathbf{b} - A\mathbf{x}_0\| \approx 1$ then $\frac{\|\mathbf{r}_k\|}{\|\mathbf{b}\|} < 10^{-m} \Rightarrow N_{histories} \approx 10^{2m}$ because of the Central Limit Theorem.

[S. Branford, C. Sahin, A. Thandavan, C. Weihrauch, V. N. Alexandrov and I. T. Dimov, 2008; P. Jakovits, I. Kromonov and S. R. Srirama, 2011]

Monte Carlo Linear Solvers

Monte Carlo Synthetic Acceleration (MCSA)

Algorithm 1: MCSA

Data: A, $\mathbf{b}$, H, $\mathbf{f}$, $\mathbf{x}_0$

Result: x_{num}

$\mathbf{x}^l = \mathbf{x}_0$;

while *not reached convergence* **do**

$\mathbf{x}^{l+\frac{1}{2}} = H\mathbf{x}^l + \mathbf{f}$;

$\mathbf{r}^{l+\frac{1}{2}} = \mathbf{b} - A\mathbf{x}^{l+\frac{1}{2}}$;

$\delta\mathbf{x}^{l+\frac{1}{2}} \approx (I - H)^{-1}\mathbf{r}^{l+\frac{1}{2}}$; "Solved" with Standard MC

$\mathbf{x}^{l+1} = \mathbf{x}^{l+\frac{1}{2}} + \delta\mathbf{x}^{l+\frac{1}{2}}$;

end

$x_{num} = x^{l+1}$

[S. Slattery, 2013 (PhD Thesis); T. M. Evans, S. R. Slattery and P. P. H. Wilson, 2013]

Adaptivity for the number of random walks

Forward Method:

$\theta_i \in \mathbb{R}$

$\theta_i = E\left[\sum_{l=0}^{\infty} W_l b_{k_l}\right]$

$\sigma_i = \sqrt{Var[\theta_i]}$

Find $\tilde{N}_i$ s.t. $\frac{\sigma_i^{\tilde{N}_i}}{|\hat{x}_i|} < \varepsilon$

$i = 1, \ldots, n$

Adjoint Method:

$\boldsymbol{\theta} \in \mathbb{R}^n$

$\boldsymbol{\theta}_i = E\left[\sum_{l=0}^{\infty} W_l d_{k_l}\right]$

$\boldsymbol{\sigma}_i = \sqrt{Var[\boldsymbol{\theta}]_i}$

Find $\tilde{N}$ s.t. $\frac{\|\boldsymbol{\sigma}^{\tilde{\boldsymbol{N}}}\|_1}{\|\hat{\mathbf{x}}\|_1} < \varepsilon$

$i = 1, \ldots, n$

Applicable preconditioning techniques

Remark: explicit knowledge of H_{ij} is needed

In fact the entry P_{ij} of the transition matrix is defined in terms of H_{ij} (Forward method) or H_{ji} (Adjoint method).

This limits the viable choices of preconditioners:

- diagonal preconditioners
- block diagonal preconditioners
- sparse approximate inverse preconditioners (AINV)

Factorized Sparse Approximate Inverse: AINV

Given a nonsingular matrix $A \in \mathbb{R}^{n\times n}$, let a_i^T and c_i^T denote the ith row of A and A^T.

Algorithm 2: AINV

Result: $Z = [z_1, z_2, \ldots, z_n]$, $W = [w_1, w_2, \ldots, w_n]$, $D = diag(p1, \ldots, p_n)$.

Let $w_i^{(0)} = z_i^0 = e_i, \quad (1 \le i \le n)$
for $i = 1, \ldots, n$ **do**
  for $j = i, i+1, \ldots, n$ **do**
    $p_j^{(i-1)} = a_i^T z_j^{(i-1)}, \qquad q_j^{(i-1)} = c_i^T w_i^{(i-1)}$
  end
  for $j = i+1, \ldots, n$ **do**

$$z_j^{(i)} = z_j^{(i-1)} - \left(\frac{p_j^{(i-1)}}{p_i^{(i-1)}}\right) z_j^{(i-1)}, \qquad w_j^{(i)} = w_j^{(i-1)} - \left(\frac{q_j^{(i-1)}}{q_i^{(i-1)}}\right) w_i^{(i-1)}$$

  end
end
Let $z_i = z_i^{(i-1)}$, $w_i = w_i^{(i-1)}$ and $p_i = p_i^{(i-1)}$, for $1 \le i \le n$.

[M. Benzi, C.D. Meyer, and M. Tuma, SISC, 1996]
[M. Benzi and M. Tuma, SISC, 1998]

Factorized Sparse Approximate Inverse: AINV

- AINV factors can be applied in different positions

$$A^{-1} \approx ZD^{-1}W^T$$

$$H = I - ZD^{-1}W^T A, \quad H = I - D^{-1}W^T AZ, \quad H = I - AZD^{-1}W^T$$

- Different ways to construct H are equivalent in terms of $\rho(H)$. However, different values of $\rho(\hat{H})$ are obtained.
- $\rho(H) < 1$ can be obtained with reasonable fill-in in many cases. On the other hand, $\rho(\hat{H}) < 1$ is more difficult to obtain.
- promising test results on parabolic problems

Monte Carlo Linear Solvers

Examples

Strictly diagonally dominant matrices (s.d.d.) with diagonal preconditioning. [A. Srinivasan, Math. Comput. Simulat. 2010]

$A \in \mathbb{R}^{n \times n}$ s.d.d. by rows: $P = diag(A)$
$H = I - P^{-1}A$, $\|H\|_\infty < 1 \quad (\Leftrightarrow \|\hat{H}\|_\infty < 1)$

$A \in \mathbb{R}^{n \times n}$ s.d.d. by columns: $P = diag(A)$
$H = I - AP^{-1}$, $\|H\|_1 < 1 \quad (\Leftrightarrow \|\hat{H}\|_1 < 1)$

Strict diag. dominance	Forward Method	Adjoint method
by rows	converges	not guaranteed
by columns	not guaranteed	converges

Monte Carlo Linear Solvers

Examples

Other examples:

- A M-matrix $\Rightarrow \exists D$ diagonal s.t. AD is s.d.d.
- with preconditioner $P = \text{block\_diag}(A)$:

$$\begin{cases} \displaystyle\sum_{\substack{j=1 \\ j\neq i}}^{p} \|A_{ii}^{-1}A_{ij}\|_\infty < 1. \quad \forall i = 1, \ldots, p \\ \\ \Rightarrow \text{ Forward method with block Jacobi converges} \end{cases}$$

$$\begin{cases} \displaystyle\sum_{\substack{i=1 \\ i\neq j}}^{p} \|A_{ii}^{-1}A_{ij}\|_1 < 1. \quad \forall j = 1, \ldots, p \\ \\ \Rightarrow \text{ Adjoint method with block Jacobi converges} \end{cases}$$

Monte Carlo Linear Solvers (MCLS)

Implementation for CPUs

Computational environment properties

- Machine employed: emmet (at ORNL)
  - number of nodes: 32
- Properties of a single CPU node
  - vendor: GenuineIntel
  - cpu_family: 6
  - model: 63
  - model name: Intel(R) Xeon(R) CPU E5-2630 v3 2.40GHz
  - cache size: 20,480 KB
  - CPU cores: 8

Monte Carlo Linear Solvers (MCLS)

Implementation on GPUs

Computational environment properties

- Machine employed: emmet (at ORNL)
  - number of devices: 4
- Properties of a single GPU device
  - vendor: NVIDIA
  - model name: Tesla K40m
  - memory clock rate: 3,004,000 KHz
  - global L1 cache supported
  - local L1 cache supported

2D parabolic problem: test case 1

Given $\Omega = (0,1) \times (0,1)$, consider

$$\begin{cases} \frac{\partial u}{\partial t} - \mu \Delta u + \boldsymbol{\beta}(\mathbf{x}) \cdot \nabla u = 0, & \mathbf{x} \in \Omega, \quad t \in (0, T] \\ u(\mathbf{x}, 0) = u_0, & \mathbf{x} \in \Omega \\ u(\mathbf{x}, t) = u_D(\mathbf{x}), & \mathbf{x} \in \partial\Omega, \quad t \in (0, T] \end{cases} \tag{7}$$

where $\mu = \frac{3}{200}$, $\boldsymbol{\beta}(\mathbf{x}) = [2y(1-x^2),\ -2x(1-y^2)]^T$,
$u_D = 0$ on $\{\{x = 0\} \times (0,1)\}, \{(0,1) \times \{y = 0\}\}, \{(0,1) \times \{y = 1\}\}$.

Discretization (`IFISS` toolbox employed):

- Quadrilateral FEM
- spatial discretization step $h = 2^{-8} \Rightarrow$ 66,049 d.o.f.'s
- time discretization step $\Delta t = 10\, h$

2D parabolic problem: test case 1

Numerical setting:

- Adjoint MCSA employed as linear solver
- relative residual tolerance: $\varepsilon_1 = 10^{-7}$
- weighted transition probability
- maximal # steps per history: 1000
- weight cut-off for history truncation: 10^{-9}
- statistical error-based adaptive parameter: $\varepsilon_2 = 0.1$
- granularity of the adaptive approach: $n_{histories} = 1000$

2D parabolic problem: test case 1

Stiffness matrix: A ($n = 66,049$). Not s.d.d.
Preconditioning: sparse factorized AINV with drop tolerance $\tau = 0.05$.
Code provided by courtesy by Miroslav Tuma. [M. Benzi and M. Tuma, SISC, 1998]

AINV preconditioner: M. Drop tolerance $\tau = 0.05$ for both factors.
Iteration matrix: $H = I - AM$.
Solution computed just for a generic time step.

- $\frac{nnz(H)}{nnz(A)} = 4.26$
- $\|H\|_1 = 1.1229 \qquad (\ \rho(H) \approx 0.9218,\ \rho(\hat{H}) \approx 0.9155\ )$
- # MCSA iterations: 6 (deterministic Richardson it.'s: 155)
- relative error$= 1.7153 \cdot 10^{-8}$
- # total random walks employed: $97,466,000$

2D parabolic problem: test case 1

CPU vs. GPU timings

- # maximal number of steps for each history fixed: $1,000$
- weight cut-off for history truncation: 10^{-9}
- # histories per numerical iteration fixed: $20,000,000$
- one CPU and one GPU nodes employed

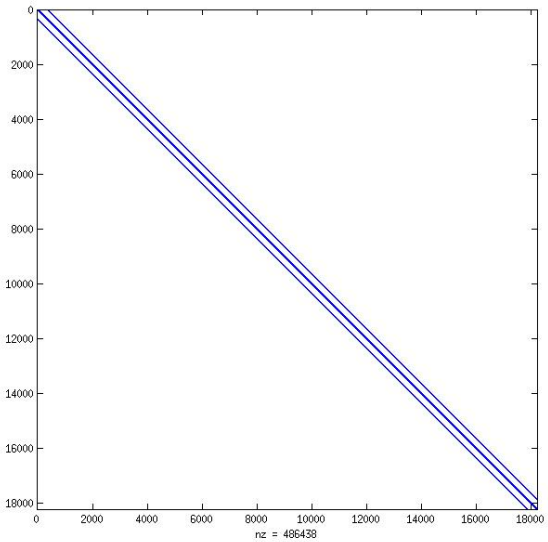
Node	Nb. Iterations	Time (sec)	Relative residual
CPU	6	992.23	$1.23 \cdot 10^{-7}$
GPU	6	29.83	$1.53 \cdot 10^{-7}$

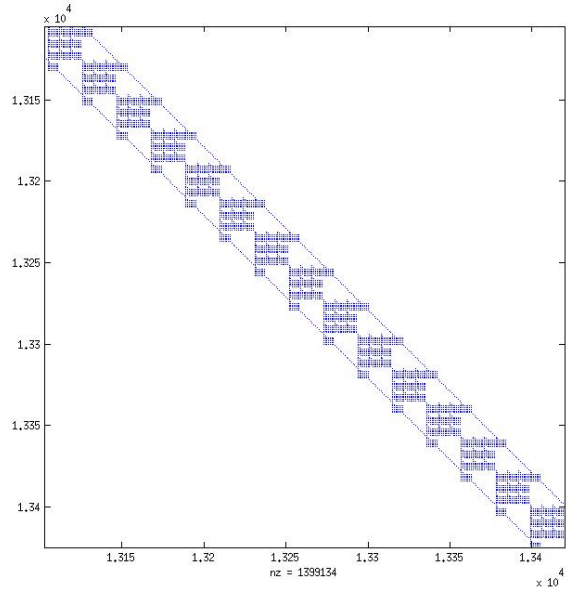
Simplified P_N equations: test case 2

Properties of the matrix:

- discretization of SP_1 equation
- all nonzero diagonal entries
- not s.d.d.

Sparsity pattern and zoom on the structure of a SP_1 matrix:





Simplified P_N equations: test case 2

Stiffness matrix: A ($n = 18,207$). Not s.d.d.
Preconditioning: block Jacobi preconditioner.
Block Jacobi preconditioner: M. Block size $p = 21$.
Iteration matrix: $H = I - AM$.

- $\frac{nnz(H)}{nnz(A)} = 2.13$
- $\|H\|_1 = 1.0066 \qquad (\ \rho(H) \approx 0.9814,\ \rho(\hat{H}) \approx 0.9813\ )$
- # MCSA iterations: 6 (deterministic Richardson it.'s: 652)
- relative error$= 3.1331 \cdot 10^{-8}$
- # total random walks employed: $64,900,000$

[Matrix provided by courtesy by Thomas Evans, Steven Hamilton, Stuart Slattery, ORNL]

Simplified P_N equations: test case 2

CPU vs. GPU timings

- # maximal number of steps for each history fixed: $1,000$
- weight cut-off for history truncation: 10^{-9}
- # histories per numerical iteration fixed: $15,000,000$
- one CPU and one GPU nodes employed

Node	Nb. Iterations	Time (sec)	Relative residual
CPU	5	$2,970$	$2.37 \cdot 10^{-7}$
GPU	5	53.2	$6.46 \cdot 10^{-7}$

Conclusions and future developments

Conclusions

- MC solvers motivated by resilience
- hypotheses for convergence difficult to satisfy in general
- currently best results obtained using AINV

Future developments

- extension of the set of matrices for which MC solvers is guaranteed to converge a priori
- analysis of how $\rho(H)$ and $\rho(\hat{H})$ affect convergence
- refinement of the adaptive selection of histories

Bibliography I


I. Dimov, V. Alexandrov, and A. Karaivanova.
Parallel resolvent Monte Carlo algorithms for linear algebra problems.
Mathematics and Computers in Simulation, 55(55):25–35, 2001.


T.M Evans, S.W. Mosher, S.R. Slattery, and S.P. Hamilton.
A Monte Carlo synthetic-acceleration method for solving the thermal radiation diffusion equation.
Journal of Computational Physics, 258(November 2013):338–358, 2014.

Bibliography II


T.M. Evans, S.R. Slattery, and P.P.H Wilson.
Mixed Monte Carlo parallel algorithms for matrix computation.
International Conference on Computational Science 2002, 2002.


T.M. Evans, S.R. Slattery, and P.P.H Wilson.
A spectral analysis of the domain decomposed Monte Carlo method for linear systems.
International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering, 2013.


J.H. Halton.
Sequential Monte Carlo.
Mathematical Proceeding of the Cambridge Philosophical Society, 58(1):57–58, 1962.

Bibliography III


J.H. Halton.
Sequential Monte Carlo techniques for the solution of linear systems.
Journal of Scientific Computing, 9(2):213–257, 1994.


J. Hao, M. Mascagni, and L. Yaohang.
Convergence analysis of Markov chain Monte Carlo linear solvers using Ulam-von Neumann algorithm.
SIAM Journal on Numerical Analysis, 51(4):2107–2122, 2013.


N. Metropolis and S. Ulam.
The Monte Carlo ethod.
Journal of th American Statistical Association, 44(247):335–341, 1949.

Bibliography IV

S. Slattery.
Parallel Monte Carlo Synthetic Acceleration Methods For Discrete Transport Problems.
PhD thesis, University of Wisconsin-Madison, 2013.

A. Srinivasan.
Monte Carlo linear solvers with non-diagonal splitting.
Mathematics and Computer in Simulation, 80:1133–1143, 2010.

Monte Carlo Linear Solvers

The choice of the transition probability

Theorem (H. Ji, M. Mascagni and Y. Li)

Let $H \in \mathbb{R}^{n \times n}$, where $\|H\|_\infty < 1$ for the Forward method and $\|H\|_1 < 1$ for the Adjoint method. Consider ν_k as the realization of a random walk γ truncated at the k-th step. Then, there always exists a transition matrix P such that $Var\left(X(\nu_k)\right) \to 0$ and $Var\left(\sum_\nu X(\nu_k)\right)$ is bounded as $k \to \infty$.

Remark If the hypotheses hold, then $P_{i,j} = \frac{|H_{i,j}|}{\sum_{k=1}^{n} |H_{i,k}|}$ (Forward) or $P_{i,j} = \frac{|H^T_{i,j}|}{\sum_{k=1}^{n} |H^T_{i,k}|}$ (Adjoint) guarantee $\|\hat{H}\|_\infty < 1$ or $\|\hat{H}\|_1 < 1$ respectively.

Monte Carlo Linear Solvers

The choice of the transition probability

Theorem (H. Ji, M. Mascagni and Y. Li)

Let $H \in \mathbb{R}^{n \times n}$ with spectral radius $\rho(H) < 1$. Let $H^+ \in \mathbb{R}^{n \times n}$ where $H^+_{i,j} = |H_{i,j}|$. Consider ν_k as the realization of a random walk γ truncated at the k-th step. If $\rho(H^+) > 1$, there does not exist a transition matrix P such that $Var\left(X(\nu_k)\right)$ converges to zero as $k \to \infty$.

Numerical experiments

Simplified P_N equations

Steady-state, multigroup, one-dimensional, eigenvalue-form of Boltzmann transport equation:

$$\mu \frac{\partial \psi^g(x,\mu)}{\partial x} + \sigma^g(x)\psi^g(x,\mu) = \sum_{g'=1}^{N_g} \int_{4\pi} \sigma_s^{gg'}(x, \hat{\Omega}\cdot\hat{\Omega}')\psi^{g'}(x,\Omega')d\Omega' + \\ + \frac{1}{k}\sum_{g'=1}^{N_g} \frac{\chi^g}{4\pi} \int_{4\pi} \nu\sigma_f^{g'}(x)\psi^{g'}(x,\Omega')d\Omega'. \tag{8}$$

$\psi^g(x,\mu)$ = angular flux for group g
σ^g = total interaction cross section
$\sigma_s^{gg'}(x, \hat{\Omega}\cdot\hat{\Omega}')$ = scattering cross section from group $g' \to g$

Numerical experiments

Simplified P_N equations

- Legendre polynomial spectral discretization (P_N equations)
- consider just odd sets of P_N equations
- removing lower order gradient terms from each equation (SP_N equations)

$$-\nabla \cdot \mathbb{D}_n \nabla \mathbb{U}_n + \sum_{m=0}^{\frac{N+1}{2}} \mathbb{A}_{nm} \mathbb{U}_m = \frac{1}{k} \sum_{m=1}^{\frac{N+1}{2}} \mathbb{F}_{nm} \mathbb{U}_m, \quad n = 1, \ldots, \frac{N+1}{2} \tag{9}$$

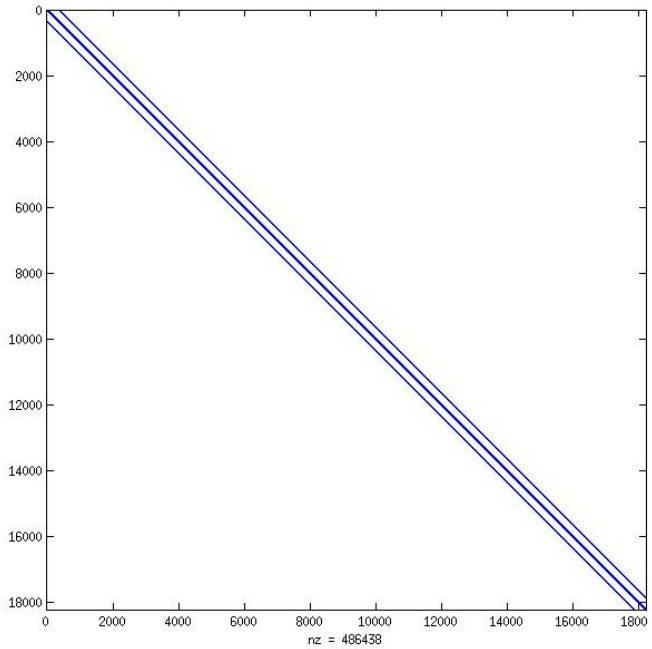
$\mathbb{U}_m$ is a linear combination of moments.

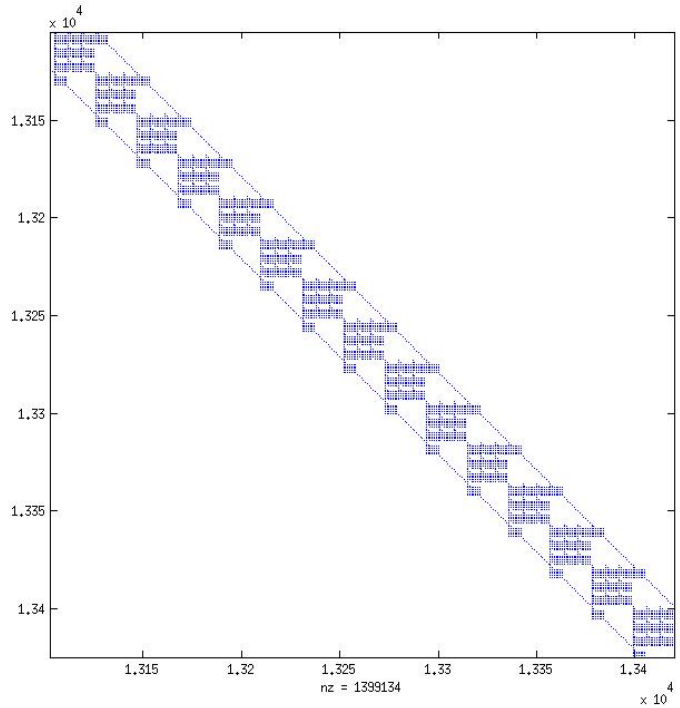
- Discretized with Finite Volumes

Numerical experiments

Simplified P_N equations

Sparsity pattern and zoom on the structure of a SP_1 matrix





Numerical experiments

Simplified P_N equations

Properties of matrices:

- discretization of SP_1 equations
- all nonzero diagonal entries

Numerical treatment:

- block Jacobi preconditioning

Kind of matrix	size	nnz	Prec. block size	nnz after prec
SP_1 (a)	18,207	486,438	63	2,644,523
SP_1 (b)	19,941	998,631	69	1,774,786

[Matrices provided by courtesy by Thomas Evans, Steven Hamilton, Stuart Slattery, ORNL]

Numerical experiments

Simplified P_N equations

MCSA parameter setting:

- Adjoint method
- residual relative tolerance: $\varepsilon_1 = 10^{-7}$
- almost optimal transition probability
- maximal # steps per history: 1000
- statistical error-based adaptive parameter: $\varepsilon_2 = 0.1$
- granularity of the adaptive approach: $n_{histories} = 1,000$
- simulations run in serial mode

Numerical experiments

Simplified P_N equations

matrix	$\rho(H)$	$\rho(\hat{H})$	relative err.	# iterations
SP_1 (a)	0.9779	0.9758	$9.97 \cdot 10^{-6}$	6
SP_1 (b)	0.9798	0.9439	$3.89 \cdot 10^{-5}$	5

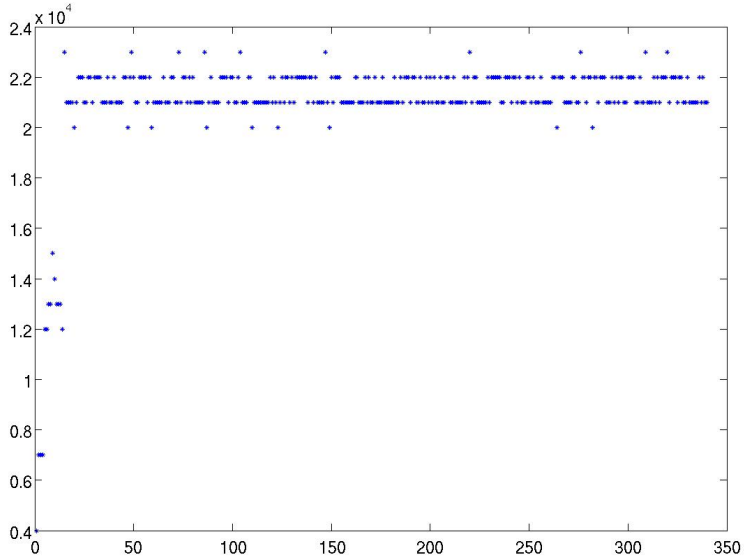


Figure 1 : # histories for (a) at each iteration.

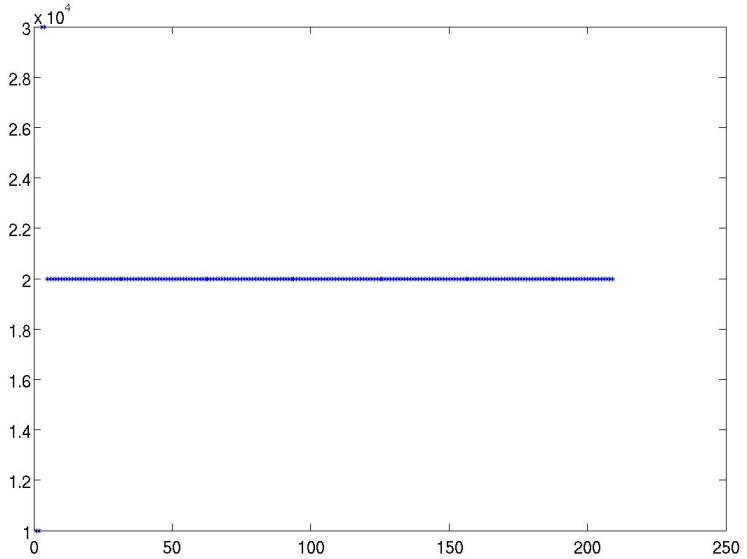


Figure 2 : # histories for (b) at each iteration.

Numerical experiments

Simplified P_N equations

Issues raised for a general SP_N matrix:

- hard to find a block Jacobi preconditioner s.t. $\rho(\hat{H}) < 1$ for every SP_N problem
- attempt to use Approximate Inverse preconditioners was not effective
  - for sparse preconditioners $\rho(\hat{H}) < 1$ is not respected
- attempt to use ILU preconditioners was not effective
  - for ILU(0) we got $\rho(\hat{H}) > 1$
  - massive fill-in with ILUT
- reordering and scaling did not facilitate the convergence requirements

Numerical experiments

Diagonally dominant matrices

Set of matrices obtained by a diagonal shift of matrices associated with SP_1, SP_3 and SP_5 equations.

$$(A + sD), \quad s \in \mathbb{R}^+, \quad D = diag(A).$$

All the matrices have been turned into s.d.d. by columns.

Initial matrix	s	$\rho(H)$	$Forward - \rho(\hat{H})$	$Adjoint - \rho(\hat{H})$
SP_1 (a)	0.3	0.7597	0.7441	0.6983
SP_1 (b)	0.4	0.7046	1.1448	0.5680
SP_3	0.9	0.5869	0.4426	0.3727
SP_5	1.6	0.5477	0.3790	0.3431

Numerical experiments

Diagonally dominant matrices

$$(A + sD), \quad s \in \mathbb{R}^+, \quad D = diag(A).$$

matrix	nnz	s	relative err.	# iterations
SP_1 (a)	486,438	0.3	-	-
SP_1 (b)	998,631	0.4	-	-
SP_3	846,549	0.9	-	-
SP_5	1,399,134	1.6	-	-

Numerical experiments

Diagonal shift

Strict diagonally dominance is a sufficient condition for convergence but not necessary.

$$(A + sD), \quad s \in \mathbb{R}^+, \quad D = diag(A)$$

such that $\rho(H) < 1$ and $\rho(\hat{H}) < 1$.

Initial matrix	s	$\rho(H)$	$Forward - \rho(\hat{H})$	$Adjoint - \rho(\hat{H})$
SP_1 (a)	0.2	0.8230	0.8733	0.8195
SP_1 (b)	0.2	0.8220	1.5582	0.7731
SP_3	0.3	0.8126	0.9459	0.7961
SP_5	0.7	0.8376	0.8865	0.8026

Numerical experiments

Diagonal shift

$$(A + sD), \quad s \in \mathbb{R}^+, \quad D = diag(A).$$

matrix	s	relative err.	# iterations
SP_1 (a)	0.2	-	-
SP_1 (b)	0.2	-	-
SP_3	0.3	-	-
SP_5	0.7	-	-