Fast Randomized Iteration: Diffusion Monte Carlo through the Lens of Numerical Linear Algebra

Raul Platero

December 1, 2017

- August 2017, Lek-Heng Lim, Jonathan Weare
- Modification to diffusion Monte Carlo techniques to solve common linear algebra problems (i.e. matrix exponentiation, solving linear systems, and eigenvalue problems)
- Fast Randomized Iteration can deal with dimensions far beyond the present limits of numerical linear algebra
- Motivated by recent application of diffusion Monte Carlo schemes to matrices as large as $10^{108} \times 10^{108}$

Diffusion Monte Carlo

- Quantum Monte Carlo: computational methods for studying quantum systems using Monte Carlo
- Diffusion is quantum but when studying zero-temperature systems
- most commonly used for computing ground state's energy of electrons (i.e. solve for the smallest eigenpair of a matrix)

Diffusion Monte Carlo

Imaginary-time Schrödinger equation:

$$\partial_t v = -\mathcal{H} v$$

Solved by the following iterative method:

$$\lambda_t = -\frac{1}{\epsilon} \log \int e^{-\epsilon \mathcal{H}} v_{t-1}(x) dx \quad \text{and} \quad v_t = \frac{e^{-\epsilon \mathcal{H}} v_{t-1}}{\int e^{-\epsilon \mathcal{H}} v_{t-1}(x) dx}$$

Include random approximations V_t^m of v_t.

$$V_t^m(x) = \sum_{j=1}^{N_t} W_t^{(j)} \, \delta_{X_t^{(j)}}(x),$$

where $\delta_y(x)$ is a Dirac delta function centered at $y \in \mathbb{R}^d$, the $W_t^{(j)}$ are real, non-negative numbers with $\mathbf{E}[\sum_{j=1}^{N_t} W_t^{(j)}] = 1$, and, for each $j \leq N_t$, $X_t^{(j)} \in \mathbb{R}^d$.

Diffusion Monte Carlo

First randomization:

$$\int f(x)[e^{\frac{\epsilon}{2}\Delta}\delta_y](x)dx = \mathbf{E}_y[f(B_\epsilon)],$$

a special case of the Feynman-Kac formula, where f is a test function, B_s is a standard Brownian motion evaluated at time $s \geq 0$.
Let

$$\tilde{V}_t^m = K_\epsilon V_{t-1}^m,$$

where K_ϵ is the discretization of $e^{-\epsilon\mathcal{H}}$. Then we can write

$$V_{t+1}^m = \frac{\tilde{V}_{t+1}^m}{\int \tilde{V}_{t+1}^m(x)dx} = \sum_{j=1}^m W_{t+1}^{(j)}\, \delta_{\xi_{t+1}^{(j)}}^{(j)},$$

where weights are recursively defined

$$W_{t+1}^{(j)} = \frac{e^{\frac{\epsilon}{2}(U(\epsilon_{t+1}^{(j)})+U(X_t^{(j)}))}W_t^{(j)}}{\sum_{\ell=1}^m e^{\frac{\epsilon}{2}(U(\epsilon_{t+1}^{(\ell)})+U(X_t^{(\ell)}))}W_t^{(\ell)}}.$$

Cost for a single iteration is $O(dm)$.

Diffusion Monte Carlo

Second randomization:

Points $\xi_j^{(j)}$ do not reference the potential U (sampled from m independent Brownian motions).

Control growth in variance by removing points with very small weights and duplicate points with large weights.
V_t^m becomes Y_t^m with

$$\mathbf{E}[Y_t^m \mid V_t^m] = V_t^m.$$

This has a cost of $O(m)$ thus overall cost per iteration is still $O(dm)$.

Diffusion Monte Carlo

1. Generate $Y_t^m = \Phi_t^m(V_t^m)$ with approximately or exactly m nonzero entries.
2. Set $V_{t+1}^m = \frac{K_\epsilon Y_t^m}{||K_\epsilon Y_t^m||_1}$.

- Unable to store iterates v_t using typical sparse matrix routines
- No direct dependence on size of K_ϵ

Fast Randomized Iteration

FRI:

$$V_{t+1}^m = \mathcal{M}(\Phi_t^m(V_t^m)),$$

where $\Phi_t^m : \mathbb{C}^n \to \mathbb{C}^n$ satisfying $\mathbf{E}[\Phi_t^m(v)] = v$.

Compression Rules

A simple choice:

$$(\Phi_t^m(v))_j = \begin{cases} N_j \frac{||v||_1}{m} \frac{v_j}{|v|_j} & \text{if } |v_j| > 0 \\ 0 & \text{if } |v_j| = 0 \end{cases}$$

where each N_j is a random, nonnegative, integer.

This scheme is suboptimal as it does not increase sparsity as error increases which drives efficiency of FRI.

Compression Rules

Algorithm 1 A simple compression rule.

Data: $v \in \mathbb{C}^n$ with all nonzero entries, $m \in \mathbb{N}$.

Result: $V = \Phi^m(v) \in \mathbb{C}^n$ with at most m nonzero entries.

$\tau_v^m = 0$;

$V = 0$;

$r = \|v\|_1/m$;

$\sigma_1 = \arg\max_i\{|v_i|\}$;

while $|v_{\sigma_{\tau_v^m+1}}| \geq r$ **do**

  $\tau_v^m = \tau_v^m + 1$;

  $V_{\sigma_{\tau_v^m}} = v_{\sigma_{\tau_v^m}}$;

  $v_{\sigma_{\tau_v^m}} = 0$;

  $r = \|v\|_1/(m - \tau_v^m)$;

  $\sigma_{\tau_v^m+1} = \arg\max_i\{|v_i|\}$;

end

For each j let N_j be a nonnegative random integer with $\mathbf{E}\,[N_j \mid v] = (m-\tau_v^m)|v_j|/\|v\|_1$. Finally, for $j \in \{1, 2, \ldots, n\} \setminus \{\sigma_1, \sigma_2, \ldots, \sigma_{\tau_v^m}\}$, set

$$V_j = N_j \frac{v_j \|v\|_1}{|v_j|(m - \tau_v^m)}.$$

(Note that v here may fewer nonzero entries than it did upon input.)

Numerical Results

Main competitor involves truncation-by-size (TbS) schemes.

TbS - thresholding where v_{σ_j} is set to zero for $j > m$ where j is largest element of v.

Results based on matrix arising in the spectral gap of a diffusion process governing the evolution of a system of up to five two-dimensional particles.

Matrix of up to size $10^{20} \times 10^{20}$

Numerical Results

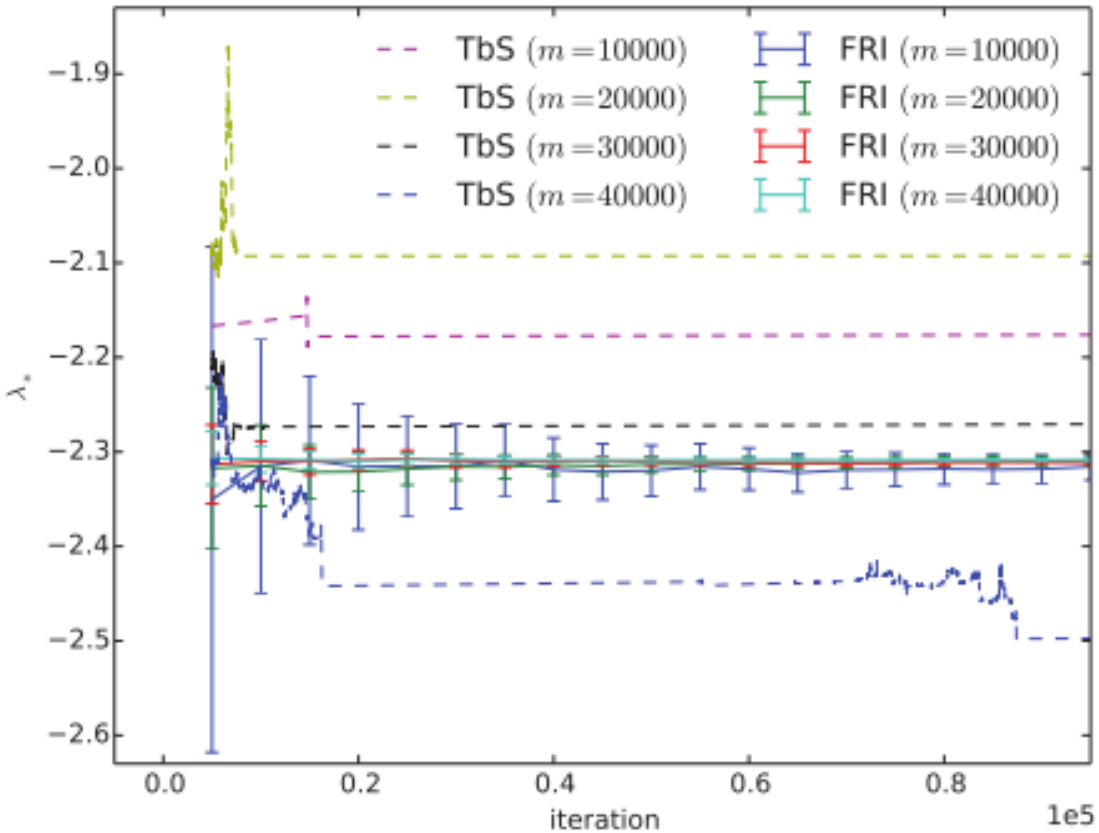

Fig. 5 *Trajectory averages of the approximation, Λ_t^m, of the largest negative eigenvalue of a backwards Kolmogorov operator for a four two-dimensional particle (eight-dimensional) system, with 95% confidence intervals for the FRI method with $m = 1, 2, 3,$ and 4×10^4. The operator is discretized using a Fourier basis with 101 modes per dimension for a total of more than 10^{16} basis elements (half that after taking advantage of the fact that the desired eigenvector is real). The step-size parameter ε is set to 10^{-3}. Also on this graph are shown trajectories of Λ_t^m for the TbS method for the same values of m.*

Numerical Results

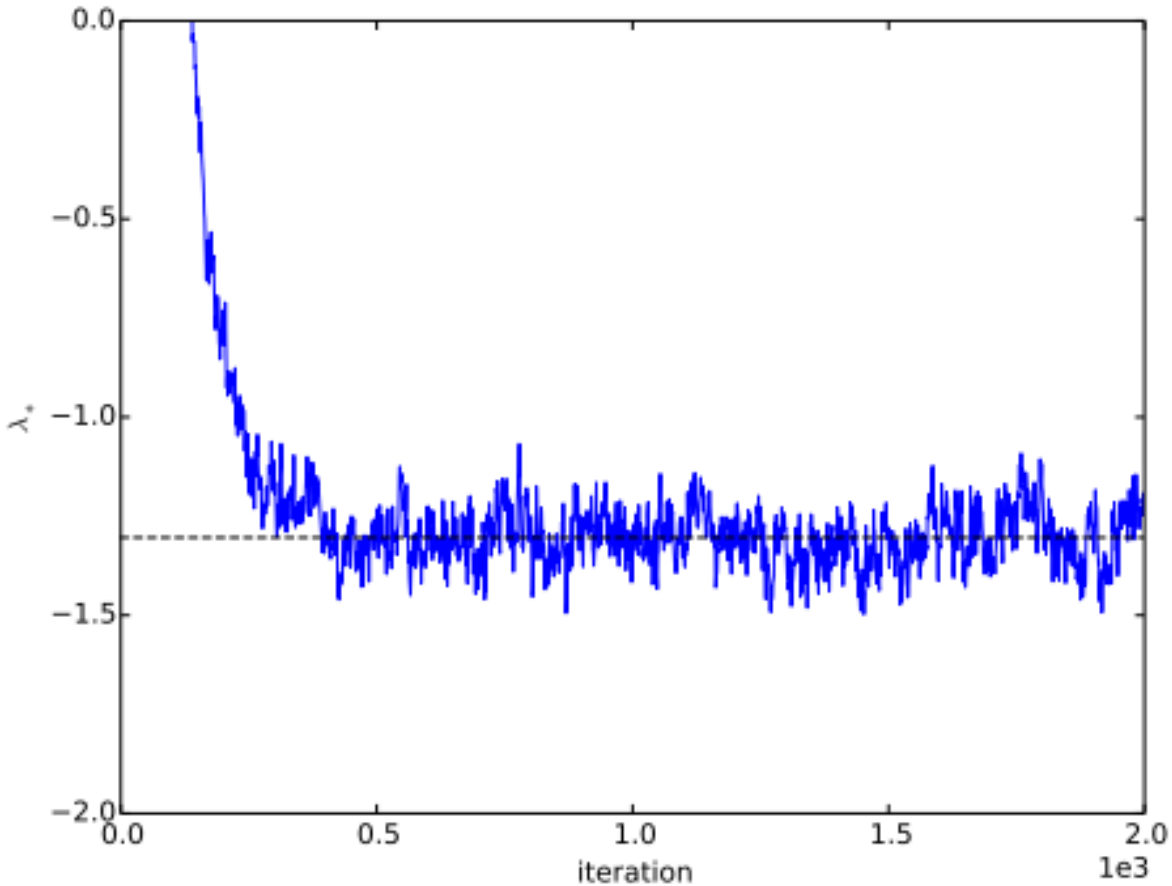

Fig. 6 *Trajectory of the approximation, Λ_t^m (solid line), of the largest negative eigenvalue of a backwards Kolmogorov operator for the five-particle system as computed by the FRI method with $m = 10^6$ over 2×10^3 iterations. The total dimension of the discretized system is more than 10^{20}. The average value of Λ_t^m (ignoring the first 500 iterations) is -1.3 and is shown by a horizontal dotted line.*