The HMC Algorithm with Overrelaxation and Adaptive–Step Discretization Numerical Experiments with Gaussian Targets

M. Alfaki, S. Subbey, and D. Haugland

Thiele Conference

July 17, 2008

Talk Outline

Bayes Theorem

- Given model–$m(\mathbf{C}):\ \mathbf{C} \in \mathcal{R}^k$, and data $\mathcal{O}$
- Bayes Theorem: Prior belief $\times$ Likelihood $\rightarrow$ Posterior

$$\frac{p(m)p(\mathcal{O}|m)}{\left[\int_{\mathcal{R}^k} p(\mathcal{O}|m)p(m)dm\right]} = p(m|\mathcal{O}). \tag{1}$$

- Posterior pdf used in inference, e.g., expectation of J: $\langle J \rangle$

$$\langle J \rangle = \int_{\mathcal{R}^k} J(m)p(m|\mathcal{O})dm = \frac{1}{n}\sum_{j=1}^{n} \frac{J(m_j)p(m_j|\mathcal{O})}{h(m_j)}, \tag{2}$$

$$\approx \frac{1}{n}\sum_{j=1}^{n} J(m_j), \text{ for } h(m_j) \approx p(m_j|\mathcal{O}). \tag{3}$$

Talk Outline

MCMC Algorithms

- Avoid calculating (intractable) integral $\int_{\mathcal{R}^k} p(\mathcal{O}|m)p(m)dm$.
- Generate ensemble of models, $m_1, m_2, \ldots, m_n | m_j \equiv m(\mathbf{C}_j)$
- Such that distribution of $\{m_j\}_{j=1}^n \sim h(m)$
- $h(m) \approx p(m|\mathcal{O}) \Rightarrow \langle J \rangle$ is an average over $m_1, m_2, \ldots, m_n$
- A popular implementation – Metropolis–Hastings algorithm
- Some example drawbacks:
  - long burn–in time
  - slow convergence (especially in high dimensions)
- Recent developments attempt to address drawbacks

Aim of Talk

Present

- The Hamiltonian Monte Carlo (HMC) Algorithm
  - A variant Monte Carlo algorithm
  - Incorporates gradient information in distribution space
- Investigated strategies for improving performance
- Numerical experimental results

Algorithm Description–I

- Type of Markov Chain Algorithm
  - Combines advantages of Hamiltonian dynamics & Metropolis MC
  - Incorporates gradients in dynamic trajectories

Given vector of parameters $\mathbf{C} \in \mathcal{R}^k$,

- Augment with conjugate momentum vector $\mathbf{P} \in \mathcal{R}^k$
- Introduce function $\mathcal{H}(\mathbf{C},\mathbf{P})$, on phase–space $(\mathbf{C},\mathbf{P})$.
- $\mathcal{H}(\mathbf{C},\mathbf{P}) \equiv$ Hamiltonian function (Classical dynamics)

$$\mathcal{H}(\mathbf{C},\mathbf{P}) = V(\mathbf{C}) + K(\mathbf{P}), \tag{4}$$

$$V(\mathbf{C}) = -\log \pi(\mathbf{C}), \quad K(\mathbf{P}) = \frac{1}{2}|\mathbf{P}|^2. \tag{5}$$

$V,\ K,\ \pi(\mathbf{C}) \equiv$ Pot. & Kinetic energies, Target distribution

Algorithm Description–II

- If $V(\mathbf{C})$ induces a Boltzmann distribution over $\mathbf{C}$

$$p(\mathbf{C}) = \frac{e^{-V(\mathbf{C})}}{\int_{\mathcal{R}^n} e^{-V(\mathbf{C})} d\mathbf{C}} \tag{6}$$

- $\mathcal{H}(\mathbf{C},\mathbf{P})$ induces a similar distribution on $(\mathbf{C},\mathbf{P})$,

$$p(\mathbf{C},\mathbf{P}) = \frac{e^{-\mathcal{H}(\mathbf{C},\mathbf{P})}}{\int_{\mathcal{R}^n}\int_{\mathcal{R}^n} e^{-\mathcal{H}(\mathbf{C},\mathbf{P})} d\mathbf{C}d\mathbf{P}} = p(\mathbf{C})p(\mathbf{P}), \tag{7}$$

$$p(\mathbf{P}) = (2\pi)^{-n/2} e^{(-\frac{1}{2}|\mathbf{P}|^2)}. \tag{8}$$

- Simulate ergodic Markov chain with stationary distrib. $\sim$ (7)
- Estimate $\langle J \rangle$– use values of $\mathbf{C}$ from successive Markov chain states with marginal distribution given by (6)

Algorithm Description–III

- Stochastic Transition
  - Draw random variable $\mathbf{P} \sim p(\mathbf{P}) = (2\pi)^{-n/2} e^{(-\frac{1}{2}|\mathbf{P}|^2)}$
- Dynamic Transition
  - New pair of $(\mathbf{C}, \mathbf{P}) \sim p(\mathbf{C}, \mathbf{P})$, starting from current $\mathbf{C}$,
  - Sample regions of constant $\mathcal{H}$ without bias
    - Ensures ergodicity of the Markov chain
- Dynamic transitions–governed by Hamiltonian equations

$$\frac{d\mathbf{C}}{d\tau} = +\frac{\partial \mathcal{H}}{\partial \mathbf{P}} = \mathbf{P}, \quad \frac{d\mathbf{P}}{d\tau} = -\frac{\partial \mathcal{H}}{\partial \mathbf{C}} = -\nabla V(\mathbf{C}). \qquad (9)$$

- Hamiltonian dynamic transitions satisfy
  - Time reversibility (invariance under $\tau \to -\tau$, $\mathbf{P} \to -\mathbf{P}$),
  - Conservation of energy ($\mathcal{H}(\mathbf{C}, \mathbf{P})$ invariant with τ)
  - Liouville's theorem (conservation of phase–space volume).

Talk Outline

Leapfrog HMC

- Choose chain length N & *leapfrog* steps L
- Simulate Hamiltonian dynamics with finite step size, ϵ.

$$\mathbf{P}(\tau + \frac{\epsilon}{2}) = \mathbf{P}(\tau) - \frac{\epsilon}{2}\nabla V(\mathbf{C}(\tau)), \tag{10}$$

$$\mathbf{C}(\tau + \epsilon) = \mathbf{C}(\tau) + \epsilon \mathbf{P}(\tau + \frac{\epsilon}{2}), \tag{11}$$

$$\mathbf{P}(\tau + \epsilon) = \mathbf{P}(\tau + \frac{\epsilon}{2}) - \frac{\epsilon}{2}\nabla V(\mathbf{C}(\tau + \epsilon)). \tag{12}$$

- Transition is volume–preserving and time–reversible
- Finite ϵ does not keep $\mathcal{H}$ constant $\rightarrow$ systematic error
- Elimate systematic error using a Metropolis rule

The Algorithm–Example Implementation

Algorithm

Initialize $C^{(0)}$

for $i = 1$ to $N-1$

Sample $u \sim U_{[0,1]}$ and $P^* \sim N(0, I)$

$C_0 = C^{(i)}$ and $P_0 = P^* + \frac{\varepsilon}{2}\nabla V(C_0)$

For $l = 1$ to L

$P_l = P_{l-1} - \frac{\varepsilon}{2}\nabla V(C_l)$

$C_l = C_{l-1} + \varepsilon P_{l-1}$

$P_l = P_{l-1} - \frac{\varepsilon}{2}\nabla V(C_l)$

end For

$dH = H(C_L, P_L) - H(C^{(i)}, P^*)$

if $u < \min\{1, \exp(-dH)\}$

$(C^{(i+1)}, P^{(i+1)}) = (C_L, P_L)$

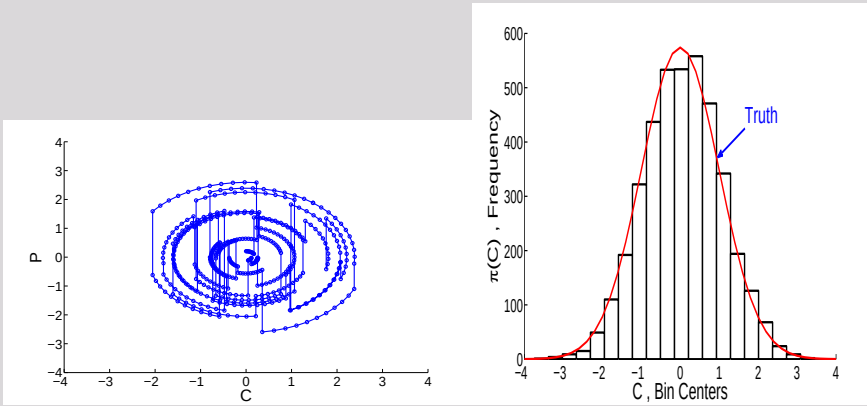
else

$(C^{(i+1)}, P^{(i+1)}) = (C^{(i)}, P^{(i)})$

end for

return $C = [C^{(1)}, C^{(2)}, \ldots, C^{(N-1)}]$

Example

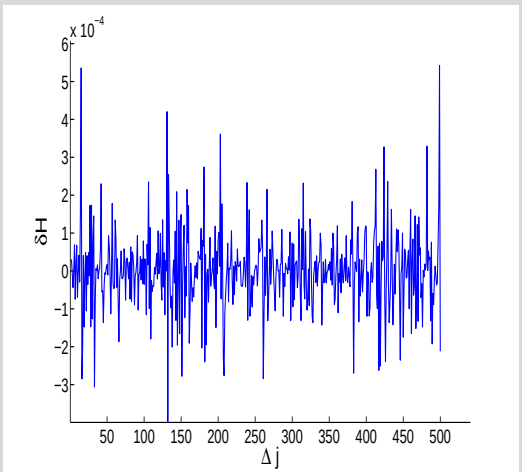


Phase-space and distribution plots for 2D correlated Gaussian distribution.

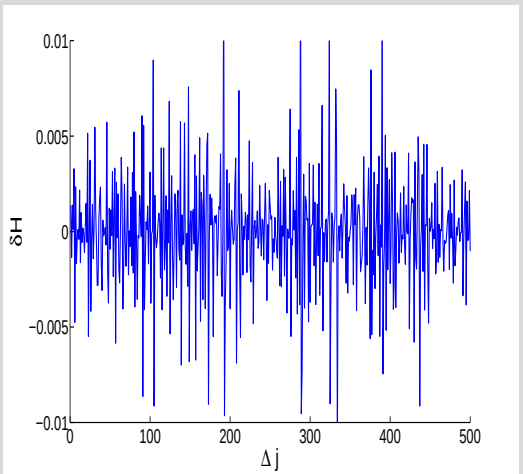
Issues with Implementation

- Given a chain of length N, the choices of L & ϵ are decisive.

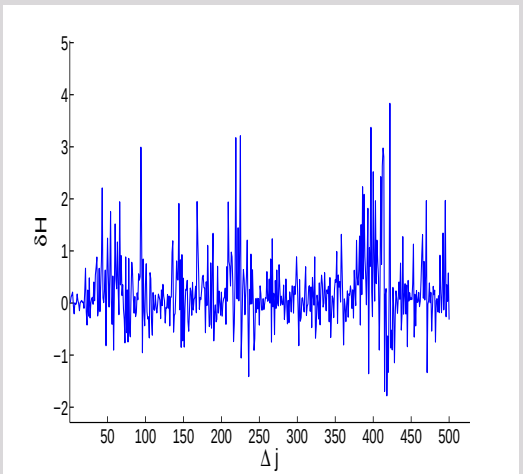
Example



(a) $\epsilon = 0.025$, $L = 20$



(b) $\epsilon = 0.1$, $L = 20$



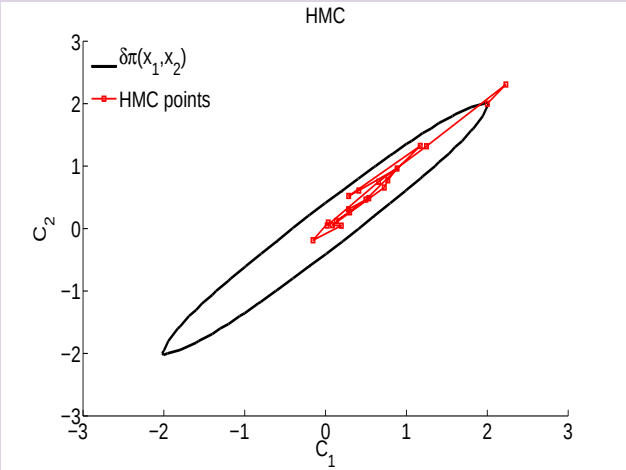
(c) $\epsilon = 1.9$, $L = 30$

Talk Outline

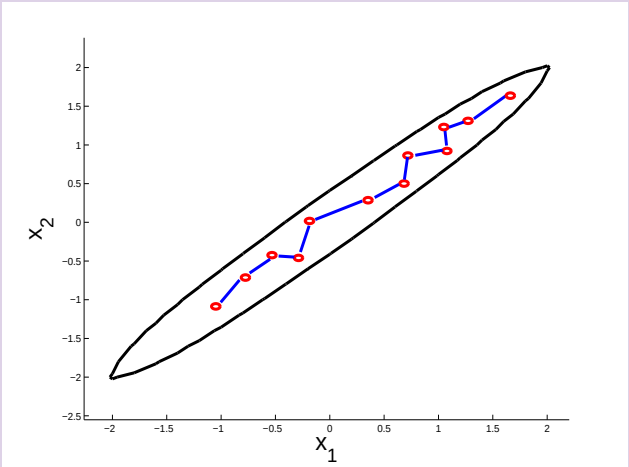
Effect of Gibbs Sampling

- Momentum variable $P \sim$ Gibbs sampler $\longrightarrow$ random walks
  - Could lead to sub–optimal sampling of phase-space
  - *Doubling on* movement leads to extra cost– CPU time

Illustration



(a) HMC walker

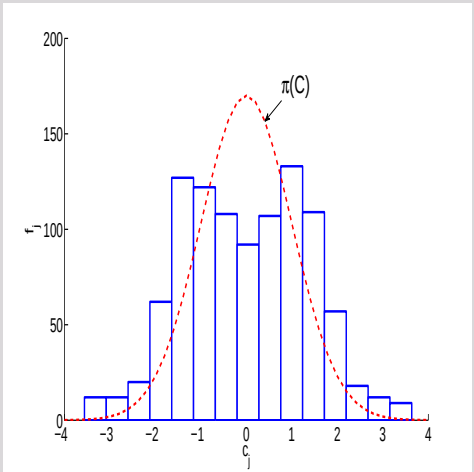


(b) Ideal walker

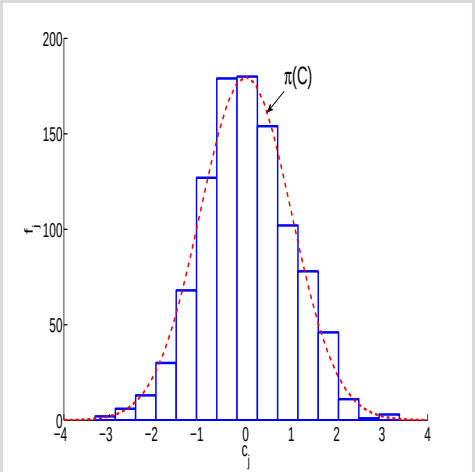
Effect of Constant Step–size

- For usual implementations, ϵ is constant
  - Inefficient when trajectory dynamics vary in different phase–space regions
  - Leads to extra cost– CPU time

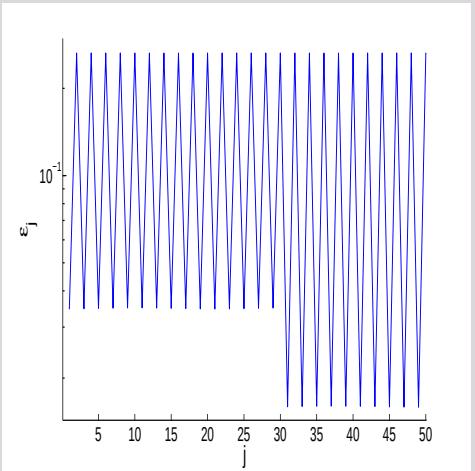
Example



(a) Constant $\epsilon = 0.1$, $L = 30$



(b) Variable ϵ, $L = 30$



(c) Variable ϵ_j

Talk Outline

Investigate Two Approaches

- **Proposal 1:** Suppressing random Walk in Gibbs sampling
  - Ordered over-relaxation (R. Neal)
- **Proposal 2:** Using a variable step–size for dynamics
  - Investigate a Runge–Kutta type integrator (simplectic)

Applying over–relaxation to P– Over-rel. HMC (OHMC)

Ordered over-relaxation

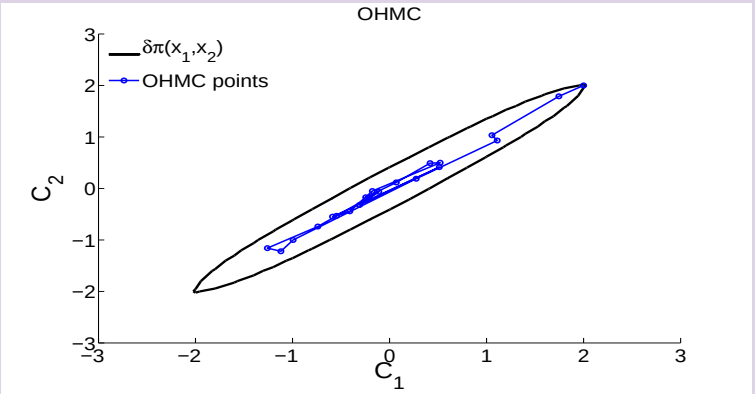
To over–relax $\mathcal{R}^n \ni P \sim \mathcal{N}(P;0,I)$

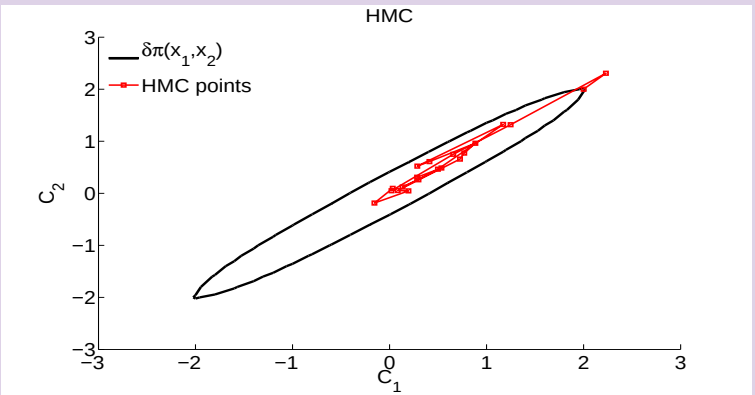
For $i = 1 : n$

- Generated K values from $\mathcal{N}(q_i|\{q_j\}_{i\neq j})$.
- Order K values and the odd value P_i. $q_i^{(0)} \leq \cdots \leq q_i^{(r)} = P_i \leq \cdots \leq q_i^{(K)}$
- Set $P_i' = q_i^{(K-r)}$.

End for

Example





Variable Step–Size HMC Algorithm (SVHMC)

- Explicit variable step–size using a Runge–Kutta scheme

Adaptive Störmer–Verlet

For $l = 1 : L - steps$

$$C_{l+\frac{1}{2}} = C_l + \frac{\epsilon}{2\rho_l} P_{l+\frac{1}{2}},$$

$$P_{l+\frac{1}{2}} = P_l - \frac{\epsilon}{2\rho_l} \nabla V(C_l),$$

$$\rho_{l+1} + \rho_l = 2U(C_{l+\frac{1}{2}}, P_{l+\frac{1}{2}}),$$

$$P_{l+1} = P_{l+\frac{1}{2}} - \frac{\epsilon}{2\rho_{l+1}} \nabla V(C_{l+1}),$$

$$C_{l+1} = C_{l+\frac{1}{2}} + \frac{\epsilon}{2\rho_{n+1}} P_{l+\frac{1}{2}}.$$

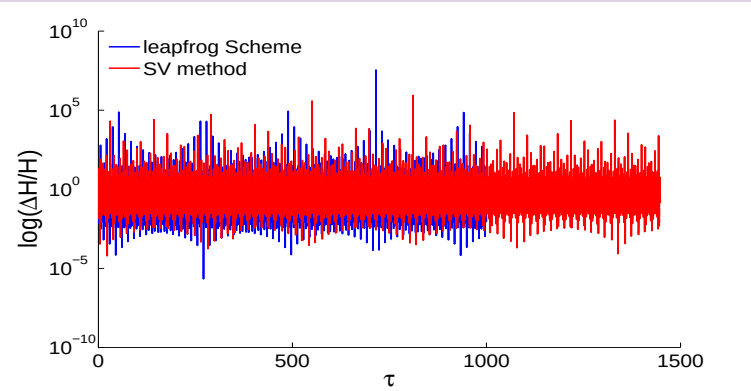
End For

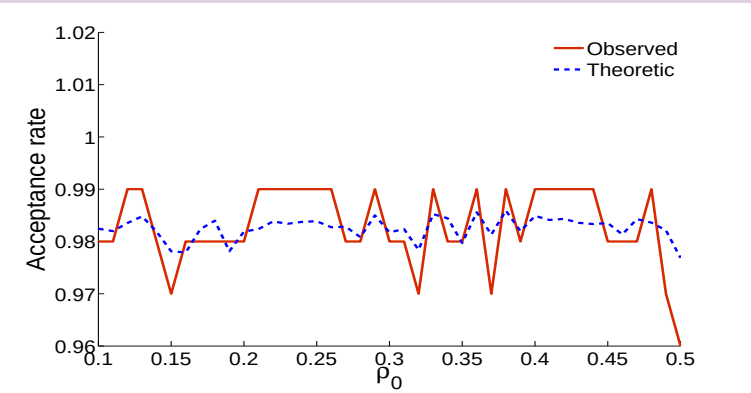
Adaptive Step–size

Adaptive Störmer–Verlet

- Adaptive ϵ reduces ΔH.
- Parameter ϵ depends on
  $U(C,P) = \sqrt{\|\nabla V(C)\|^2 + P^T[\nabla^2 V(C)]^2 P}$
- Observed $\sim$ theoretical acceptance rates
- ρ_o is a fictive parameter

Example





Numerical Experiments

- Gaussian targets with uncorrelated covariates in 64 & 128D

$$\pi(\mathbf{C}) = \frac{1}{(2\pi)^{\frac{D}{2}} \det(\Sigma)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\mathbf{C}^T\Sigma^{-1}\mathbf{C}\right). \qquad (13)$$

- Compare HMC, SVHMC & OSVHMC algorithms based on
  - Degree of chain autocorrelation
  - Effective number of samples in a given chain
  - Variance of sample means, $\overline{\mathbf{C}}$, of a finite chain
  - Convergence rates/ratio
  - Dimensionless efficiency,

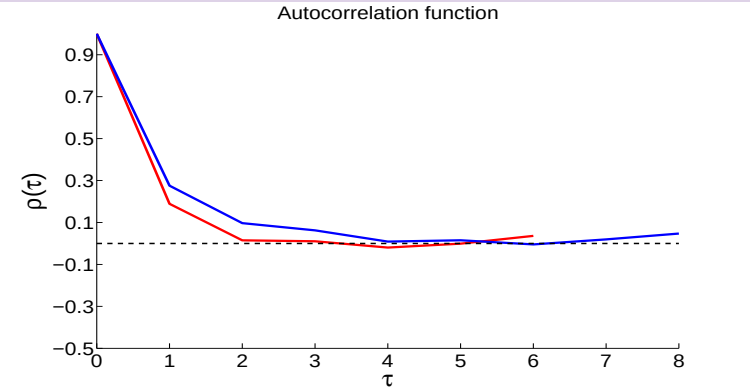
Evaluation Criteria

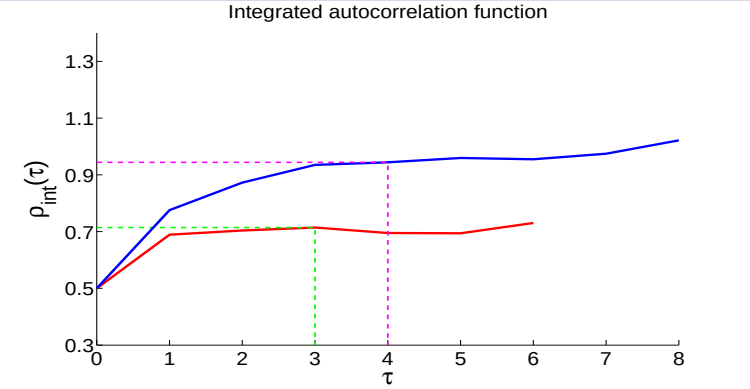
Suppose $\{c_i\}_{i=1}^{N}$ is chain generated by algorithm.

1. Degree of correlation criteria
   - Autocorrelation function $\rho(l) = \frac{Cov(x_i, x_{i+l})}{Var(x_i)}$
   - Integrated autocorrelation time $\tau_{int} = \frac{1}{2} + \sum_{t=1}^{\infty} \rho(t)$
   - Effective sample size $N_{eff} = N/(2\tau_{int})$
2. Spectral analysis criteria
   - Compute $\tilde{P}_j = |\tilde{C}(\kappa)^* \tilde{C}(\kappa)|, \ \tilde{C}(\kappa) = \mathrm{DFT}(c)$
   - Fit template $P(\kappa) = P_0 \frac{(\kappa^*/\kappa)^{\alpha}}{(\kappa^*/\kappa)^{\alpha}+1}$ to $\tilde{P}_j$
     - α, $P(0)$ & κ^* – parameters to be estimated
   - The sample mean variance $\sigma_{\bar{x}}^2 \approx P(\kappa = 0)/N$
   - Convergence ratio $r = \sigma_{\bar{x}}^2/\sigma_0^2$
   - The dimensionless efficiency $E = \lim_{N\to\infty} \frac{\sigma_0^2/N}{\sigma_{\bar{x}}^2(N)}$

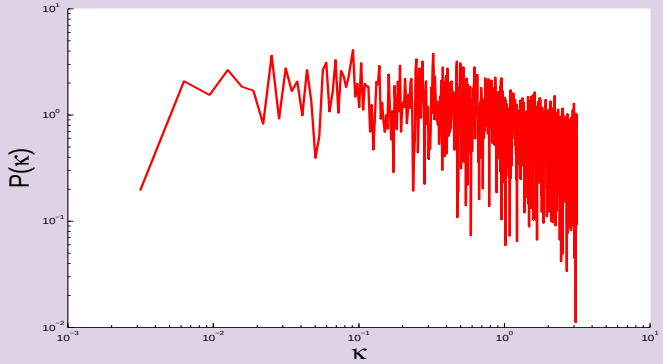
Evaluation criteria – Geometric Illustration

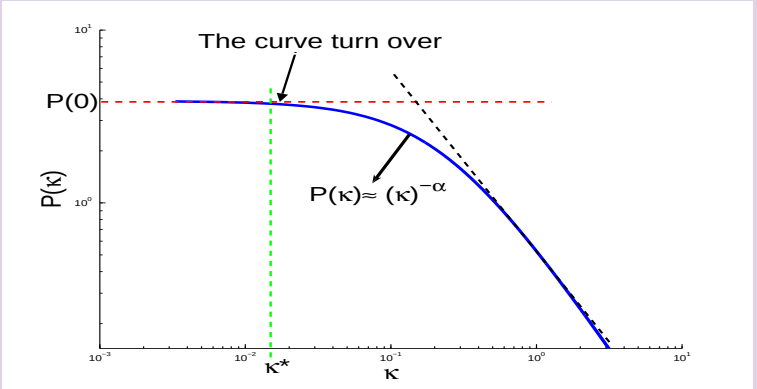
Degree of correlation





Spectral analysis





Talk Outline

Comparing OHMC vs HMC

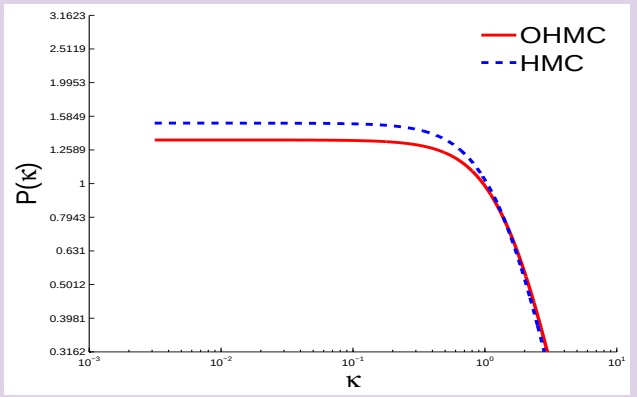
Gaussian Target

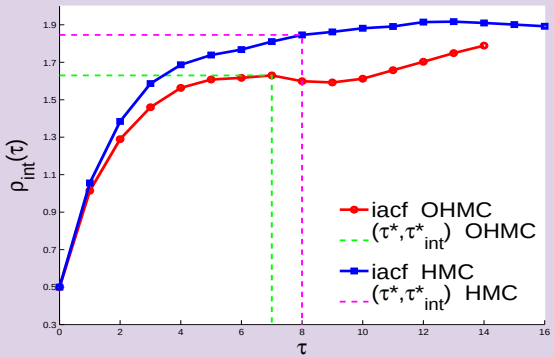
$$\pi(x) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}x^T\Sigma^{-1}x\right)$$
$$\Sigma = I$$

Results (n=64, N=2000)

	OHMC	HMC	Ideal
Accept. rate	0.99	0.99	1
$P(0)$	1.35	1.51	1
κ^*	1.65	1.45	
CPU time[sec]	561.22	557.38	
E	0.74	0.66	1
r	$6.7e-4$	$7.6e-4$	< 0.01
τ_{int}	1.63	1.85	0.5
N_{eff}	614	542	2000

Graphical Illustration



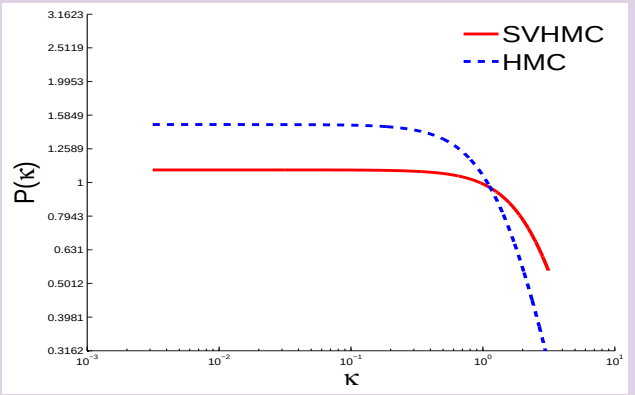


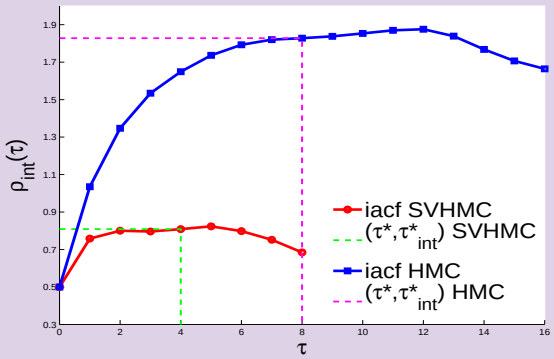
Comparing SVHMC vs HMC

Numerical Results (n=128 N=2000)

	SVHMC	HMC	Ideal
Accept. rate	0.92	0.98	1
$P(0)$	1.09	3.13	1
κ^*	3.15	1.55	
CPU time[sec]	1568.01	1117.78	
E	0.92	0.67	1
r	$5.6e-4$	$7.4e-4$	< 0.01
τ_{int}	0.86	1.80	0.5
N_{eff}	1167	554	2000

Graphical Illustration



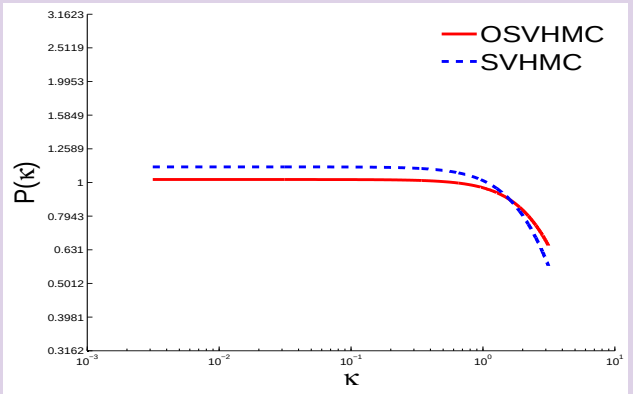


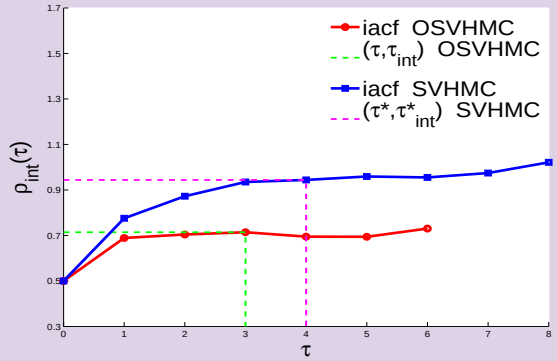
Comparing OSVHMC vs SVHMC

Numerical Results (n=64, N=2000)

	OSVHMC	SVHMC	Ideal
Accept. rate	0.92	0.94	1
$P(0)$	1.02	1.11	1
κ^*	4.15	3.19	
CPU time[sec]	639.58	669.40	
E	0.98	0.90	1
r	$5.1e-4$	$5.6e-4$	< 0.01
τ_{int}	0.71	0.94	0.5
N_{eff}	1400	1059	2000

Graphical Illustration





Summary and Conclusion

1. Over-relaxation in the Gibbs sampling improves dimensionless efficiency by a factor $\sim 12\%$.

$$\frac{E_{OHMC}}{E_{HMC}} \approx 1.2$$

2. Using Störmer–Verlet discretization outperforms the leapfrog HMC by having $\sim 50\%$ more effective sample size

$$\frac{N_{eff}^{SV}}{N_{eff}^{leapfrog}} \approx 2.0$$

3. The hybrid– OSVHMC (over-relaxing the momentum & Adaptive ϵ) outperform the SVHMC