

Hamiltonian Monte Carlo in ADMB and TMB: current status and future directions

Cole Monnahan

ADMB/TMB Developers' Workshop

6/23/2016

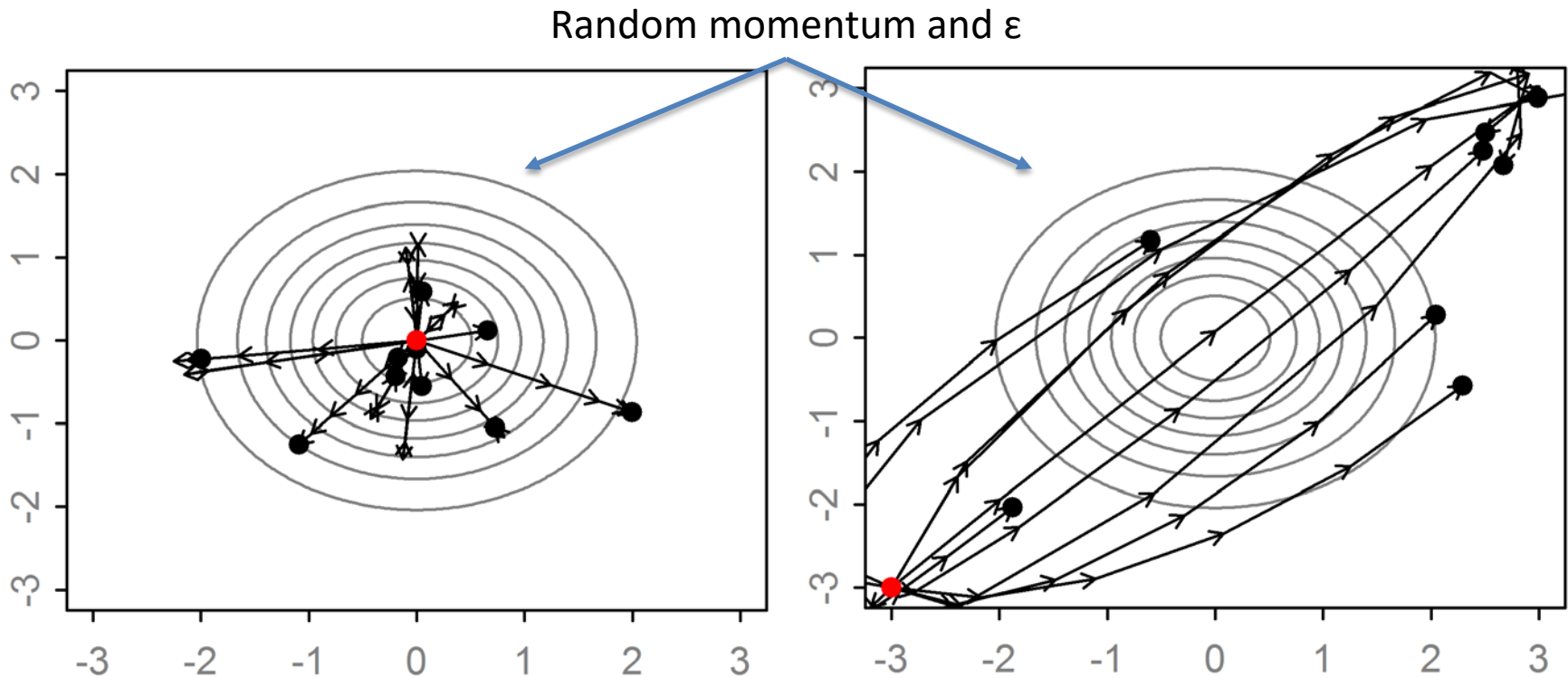
Road map

- What is HMC? (8 min)
- Stan demo (5 min)
- Status in ADMB, TMB, and Stan (10 min)
- HMC extensions (2)
- Future development: parameter bounds, RMHMC, UI, error handling, etc. (5 min)
- Discussion (30)

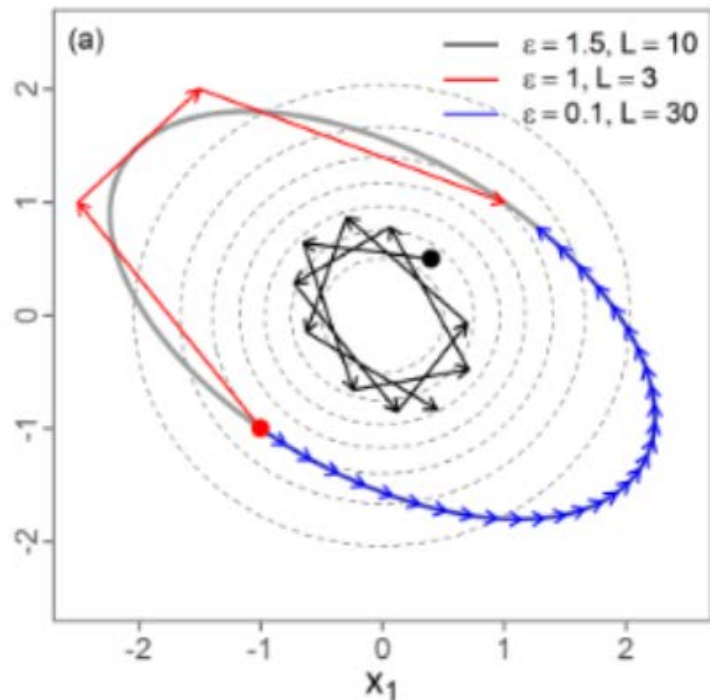
Hamiltonian Monte Carlo

- Family of MCMC algorithms which uses gradients
- Scales well with dimensionality, complexity, and hierarchical structure
- Original algorithm (hybrid or static HMC) is sensitive to tuning parameters

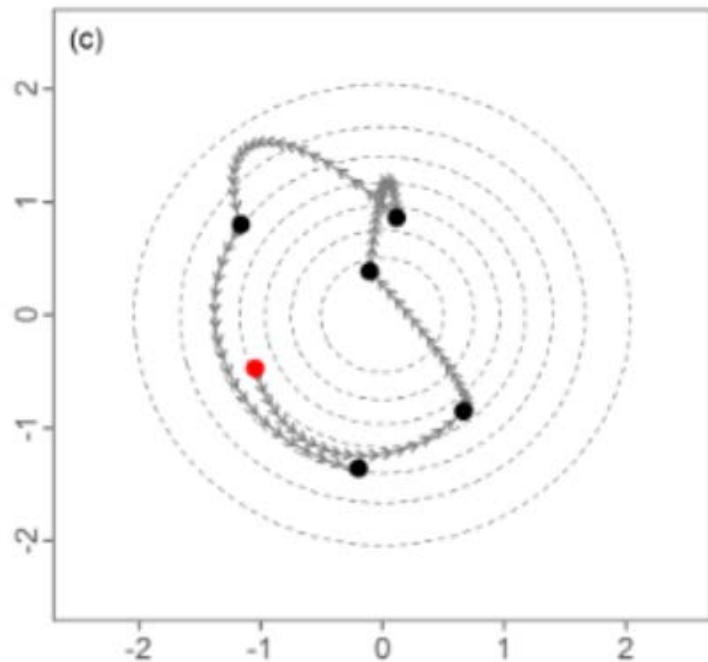
Static HMC trajectories



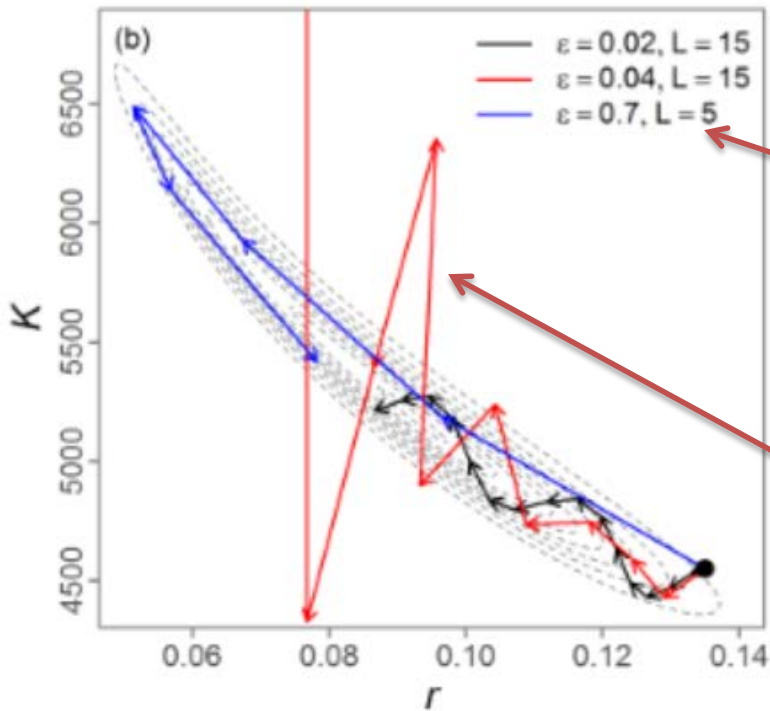
- Trajectories are approximations of paths in finite time.
- Used to propose transitions between distant areas
- Avoid random walk behavior due to directed movement.



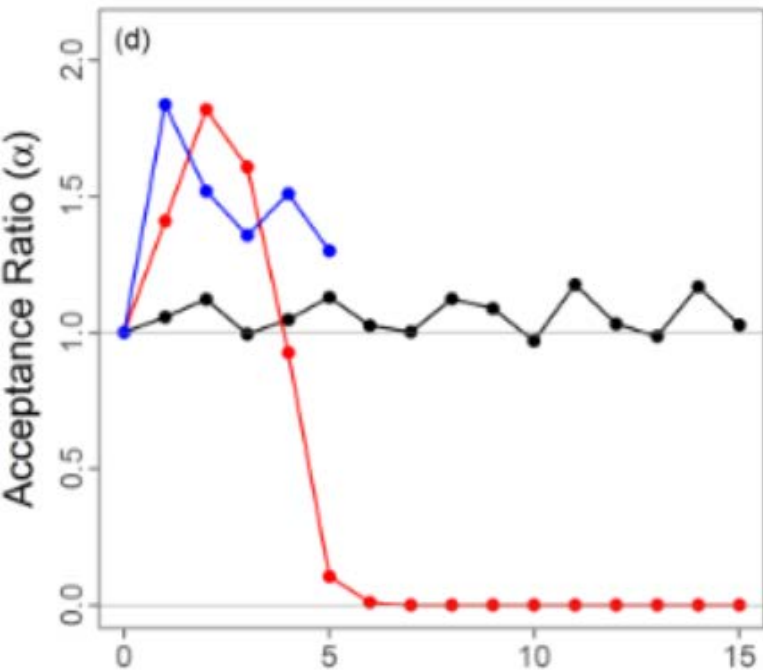
- Uses leapfrog integrator to propose states:
 - Stepsize (ϵ)
 - # steps (L)



- Intermediate steps discarded



- Mass matrix
 - Stretches and rotates posterior
- Divergence
 - Approximation errors accumulate



- Acceptance Ratio
 - Accounts for errors in H

No-U-Turn Sampler

- **Extend static HMC to avoid specifying L and ϵ .**
- ϵ is adapted with ‘dual averaging’. (Works for static HMC too).
- L is set automatically by repeatedly doubling until a “U-turn” occurs.
- L varies at each transition, avoids wasteful steps.
- Only one tuning parameter: target acceptance rate (0.6-0.9)

No-U-Turn Trajectory

for j in $0:\text{max_depth}$

Pick random direction (left or right)

Recursively build tree of size 2^j

If U-turn occurs in subtree or divergence
break, excluding subtree

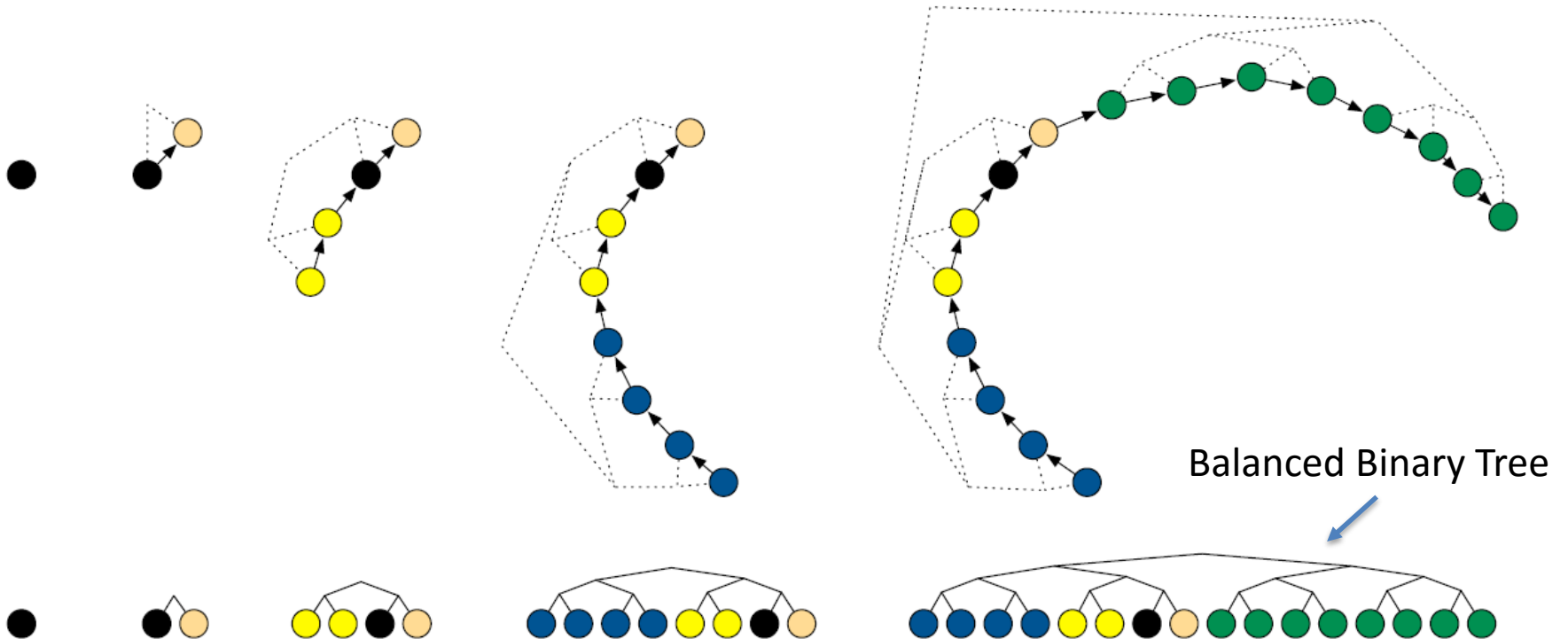





Fig 1, Hoffman and Gelman (2014)

Stan template: mixed effects model

```
data {  
  int<lower=0> J; // number of schools  
  real y[J]; // estimated treatment effects  
  real<lower=0> sigma[J]; // s.e. of effect estimates  
}  
parameters {  ADMB-like sections  
  real mu;  
  real<lower=0> tau;  
  real eta[J];  
}  
transformed parameters {  
  real theta[J];  
  for (j in 1:J)  Noncentered random effects  
    theta[j] <- mu + tau * eta[j];  
}  
model {  BUGS-like distribution declarations  
  eta ~ normal(0, 1);  
  y ~ normal(theta, sigma);  
}
```

rstan interface

```
fit <- stan(file='8schools.stan', data=schools_dat,  
            iter=2000, chains=4,  
            control=list(adapt_delta=.8))
```

← NUTS default

↖ Target acceptance rate

```
> str(fit)  
Formal class 'stanfit' [package "rstan"] with 10 slots  
 ..@ model_name: chr "8schools"  
 ..@ model_pars: chr [1:5] "mu" "tau" "eta" "theta" ...  
 ..@ par_dims  :List of 5  
 .. ..$ mu      : num(0)  
 .. ..$ tau      : num(0)  
 .. ..$ eta      : num 8  
 .. ..$ theta    : num 8  
 .. ..$ lp__     : num(0)  
 ..@ mode       : int 0  
 ..@ sim        :List of 12  
... [truncated].
```

Stan ecosystem

- `stan`: Main code base (algorithms, etc.)
- `cmdstan`, `rstan`, `pystan`, `statastan`:
interfaces for command line, R, Python and Stata.
Contain tools for diagnostics, plotting, etc.
- `rstanarm`: Similar to `glmmTMB`
- `shinystan`: Analysis & visualization GUI
for MCMC ([demo](#))

Current status of HMC

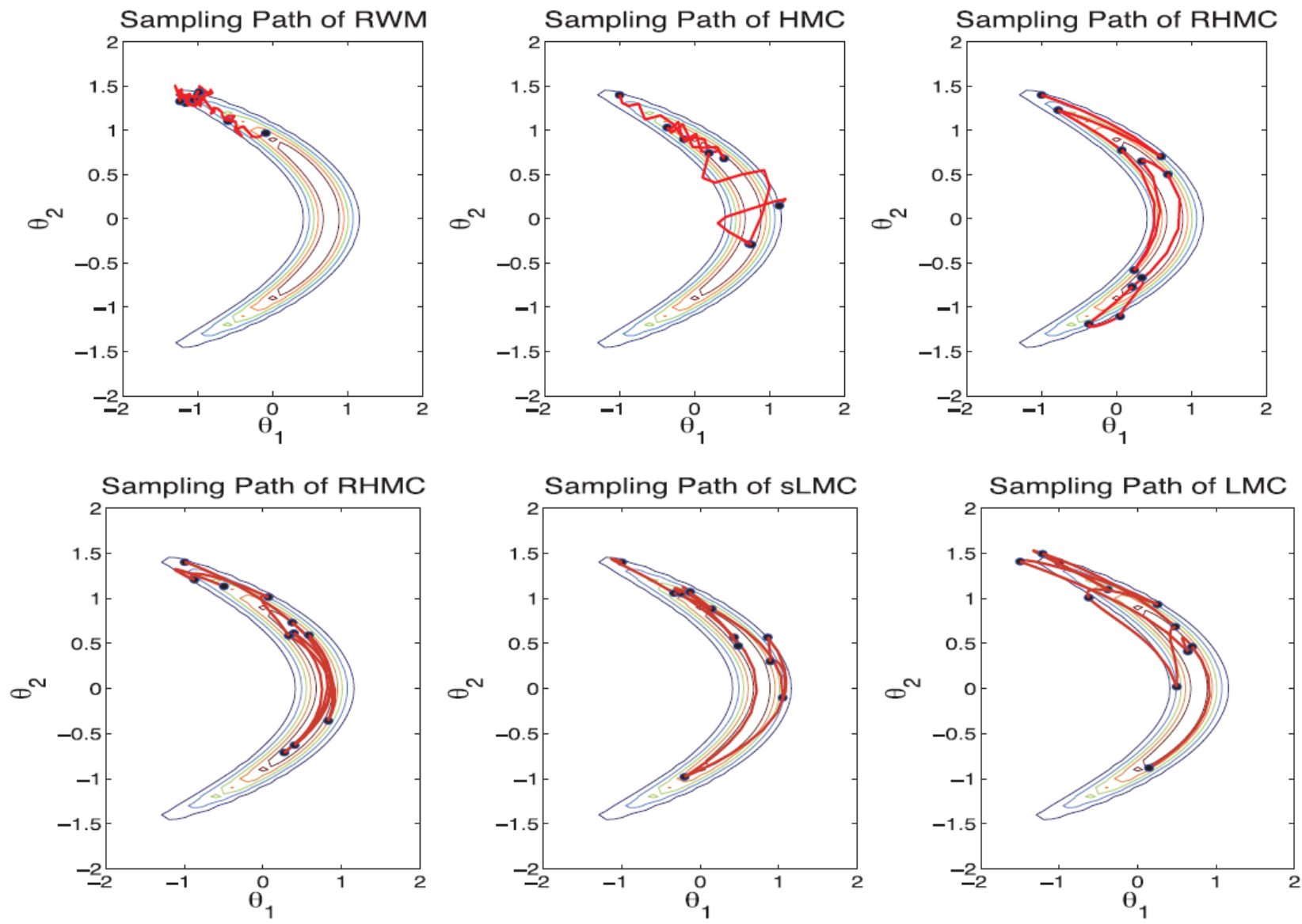
	ADMB	TMB	Stan
Static HMC	✓	✓	✓
NUTS	✗	✓	✓
Dual averaging	✗	✓	✓
ADVI	✗	✗	✓
RMHMC	✗	✗	Development?
Parallel chains	R hack	Template or R side	R side
Mass matrix	Estimated covariance; No adaptation	Arbitrary matrix; No adaptation	Unit diagonal; adapted diagonal/dense

HMC Extensions

- Riemann Manifold HMC (Girolami & Calderhead, 2011)
 - Mass matrix varies across posterior
- Lagrangian HMC (Lan et al., 2014)
 - Replaces Hamiltonian with Lagrangian dynamics (velocity instead of momentum)
- Adiabatic HMC (Betancourt 2015)
 - ‘...powerful approach to exploring the complex and multimodal target distributions...’
- Improved adaptation schemes? (Wang et al., 2013)
- Bottom line: **HMC is evolving quickly**

Extension trajectories

Lan et al. 2014



Future work

1. Add NUTS and dual averaging (ADMB)
2. Arbitrary mass matrices (ADMB)
3. Add parameter bounds (TMB)
 - Currently done via R interface
4. Robustness and thorough testing (both)
 - Utilize Stan models/tests
 - Step size converging to same values?
 - $\min(\text{ESS})$ equivalent?

Discussion points

- Piggyback on Stan (theory, testing, code)?
 - Argument names, console & output format, etc.
- Parameter bounds for TMB
- Mass matrix adaptation? From MLE?
- Add RMHMC? ADVI?
- Parallelization
- MCMC for FE; LA for RE ('-mcmc2')
- mcmc_phase() for TMB?