ILMATIETEEN LAITOS
METEOROLOGISKA INSTITUTET
FINNISH METEOROLOGICAL INSTITUTE

Efficient parameter estimation with the MCMC toolbox

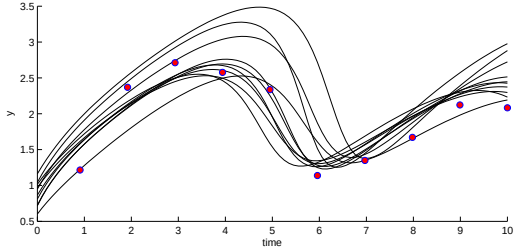
Marko Laine
marko.laine@fmi.fi

Finnish Meteorological Institute

DTU – MCMC lectures, part II – 17.12.2018

Statistical analysis by simulation

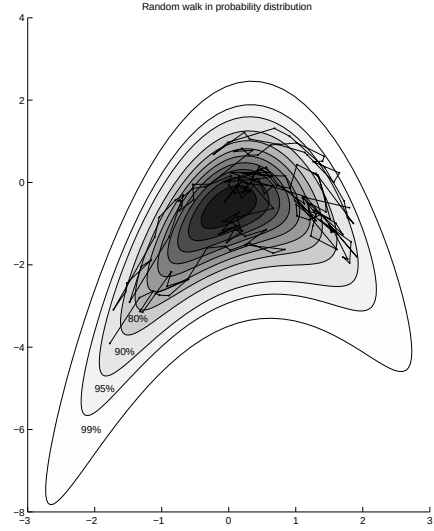
- We are interested in the uncertainty distribution of the unknown model parameter vector θ given the observational data y and the model: $p(\theta|y, M)$.
- This distribution is typically analytically intractable.
- We can still simulate observations / realizations from $p(y|\theta, M)$.



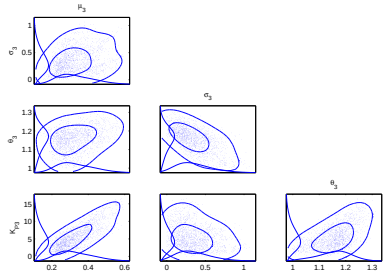
- Statistical inference is used to define what is a good fit. Parameters that are consistent with the data and the modelling uncertainty are accepted.

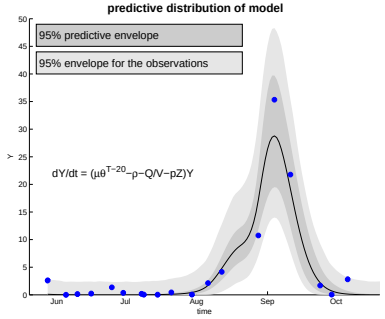
MCMC - Markov chain Monte Carlo

- Simulate the model while sampling the parameters from a *proposal distribution*.
- Weight the (or accept) the parameters according to a suitable goodness-of-fit criteria depending on *prior information* and error statistics.
- The resulting chain (i.e. a sample of parameters) is a sample from the Bayesian *posterior distribution* of parameter uncertainty.



Posterior distributions





- Posterior distribution of model parameters.
- Posterior distribution of model predictions.
- Posterior distribution for model comparison.

Terminology

The model in general form is

$$y = f(x|\theta) + \epsilon,$$
$$\text{observations} = \text{model} + \text{error}.$$

Likelihood function for Gaussian errors corresponds to a quadratic cost function, with

$$p(y|\theta) \propto \exp\left\{-\frac{1}{2}\frac{\sum_i^n \left(y_i - f(x_i|\theta)\right)^2}{\sigma^2}\right\}$$
$$= \exp\left\{-\frac{1}{2}\frac{SS(\theta)}{\sigma^2}\right\},$$

where $SS(\theta) = -2\log(p(y|\theta))$ is the -2 times log-likelihood in "sum-of-squares" cost function format. For the posterior, we also need to account $SS_{\text{pri}}(\theta) = -2\log(p(\theta))$, prior "sum-of-squares".

Metropolis-Hastings algorithm

Random walk Metropolis-Hastings algorithm with Gaussian proposal distribution (and Gaussian likelihood).

- Propose new parameter value $\theta_{\text{prop}} = \theta_{\text{curr}} + \xi$, where $\xi \sim N(0, \Sigma_{\text{prop}})$ is drawn from the proposal distribution.
- Accept θ_{prop} with probability α,

$$\alpha(\theta_{\text{curr}}, \theta_{\text{prop}}) = 1 \wedge \exp\left\{ -\frac{1}{2}\left(\frac{SS(\theta_{\text{prop}}) - SS(\theta_{\text{curr}})}{\sigma^2} \right) - \frac{1}{2}\left(SS_{\text{pri}}(\theta_{\text{prop}}) - SS_{\text{pri}}(\theta_{\text{curr}}) \right) \right\}$$

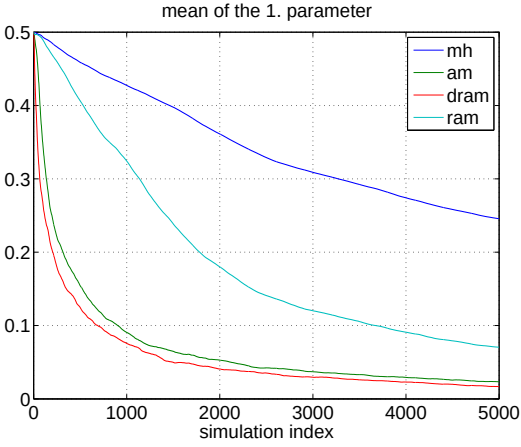
- Efficient proposal distribution $\Rightarrow$ adaptive tuning of Σ_{prop}, AM, DRAM algorithms.

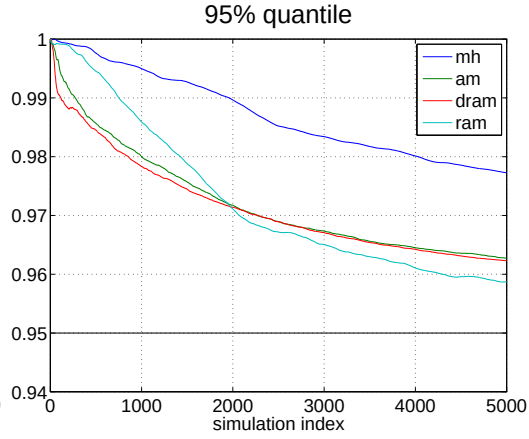
Convergence diagnostics

- 1d and 2d plots of the chain
- Mixing, rejection rate.
- Monte Carlo error of the estimates, chain autocorrelation.
- Efficient number of simulations, integrated autocorrelation time, batch mean standard error.

Short chains and adaptation

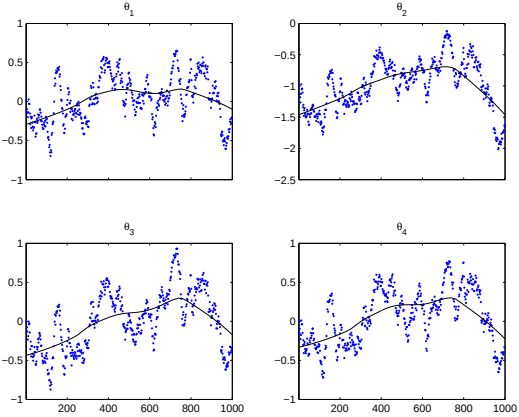
- Important to make short chains as efficient as possible.
- Efficient: produce estimates with small Monte Carlo error.



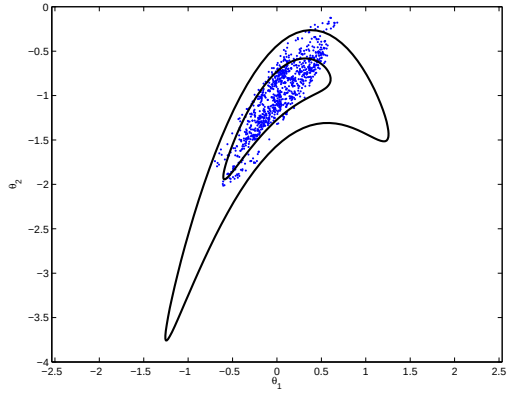


Short MCMC chain run 1000 times with different algorithms.

Chains have not mixed yet



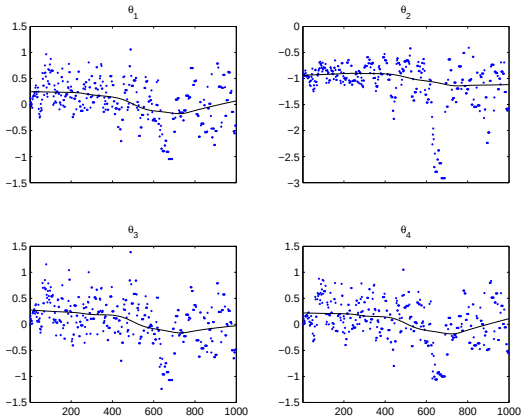
2 first dimensions of the chain with 50% and 95% target contours
c50 = 78.7%, c95 = 98.8%, τ=95.8



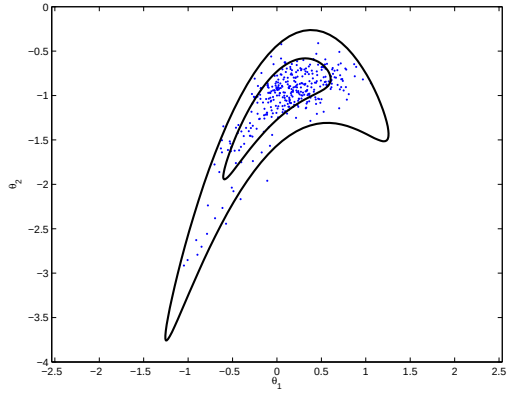

```
>> chainstats(chain,names)
MCMC statistics, nsimu = 1000
```

	mean	std	MC_err	tau	geweke
theta_1	0.037146	0.29164	0.054374	67.086	0.017371
theta_2	-0.98318	0.39323	0.082073	112.74	0.43098
theta_3	0.0077533	0.37183	0.076218	101.3	0.013535
theta_4	0.058239	0.34665	0.069839	101.88	0.014742

Chains that have better mixing



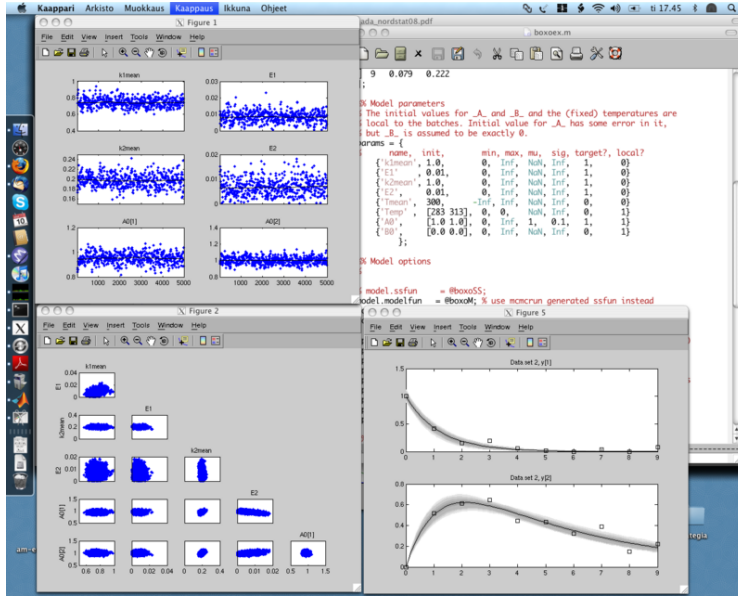
2 first dimensions of the chain with 50% and 95% target contours
c50 = 58.8%, c95 = 98.3%, τ=44.0




```
>> chainstats(chain,names)
MCMC statistics, nsimu = 1000
```

	mean	std	MC_err	tau	geweke
theta_1	0.047117	0.42071	0.068536	43.274	0.027867
theta_2	-1.1139	0.49858	0.088196	47.542	0.60531
theta_3	0.050454	0.43527	0.069026	41.525	0.023323
theta_4	0.033922	0.42226	0.067013	43.516	0.038749

MCMC Toolbox for Matlab



https://github.com/mjlaine/mcmcstat

MCMC toolbox for matlab

```
%% MCMC toolbox
model.ssfun = @mycostfun

data = load('datafile.dat');

parameters = {
  {'par1', 2.3 }
  {'par2', 1.2 }
};

options.nsimu  = 5000;
options.method = 'am';

[results,chain] = mcmcrun(model,data,parameters,options);

mcmcplot(chain,[],results)
```

MCMC demos

Matlab source code: `https://github.com/mjlaine/mcmcstat`
Documentation: `https://mjlaine.github.io/mcmcstat/`
Installation by git:

```
git clone https://github.com/mjlaine/mcmcstat.git
```