Numerical Cosmology 4

MCMC Tutorial

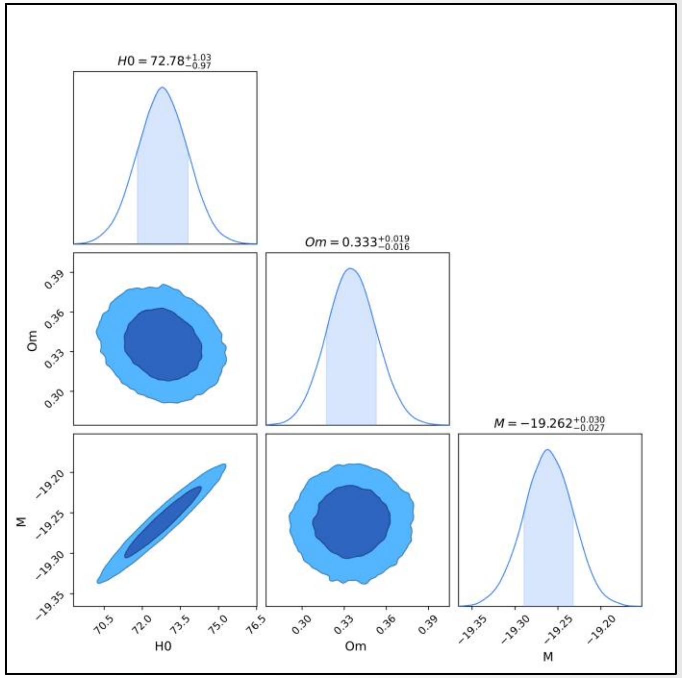
Jackson Levi Said

Institute of Space Sciences and Astronomy, University of Malta

Tutorial Plan

- Download likelihoods and CC and SN datasets from
  bit.ly/4kXAPeW

- In *main*, write
  - log_prior ($50 < H0 < 100, 0.0 < \Omega_{m0} < 0.5, -50 < M < 50$)
  - log_probability (CC_Cov().loglkl(theta) & Likelihood_Pantheon_SH0ES().loglkl(theta))
  - Set 50 walkers
  - Set 5000 Iterations
  - Plot output using example ChainConsumer code



- In *Solver*, write
  - Friedmann equation ($H(z) = H_0\sqrt{\Omega_{m0}(1+z)^3 + (1-\Omega_{m0})}$)
  - Use example code for the angular diameter distance ($d_A = \frac{c}{1+z}\int \frac{dz}{H(z)}$) or write new integrator code

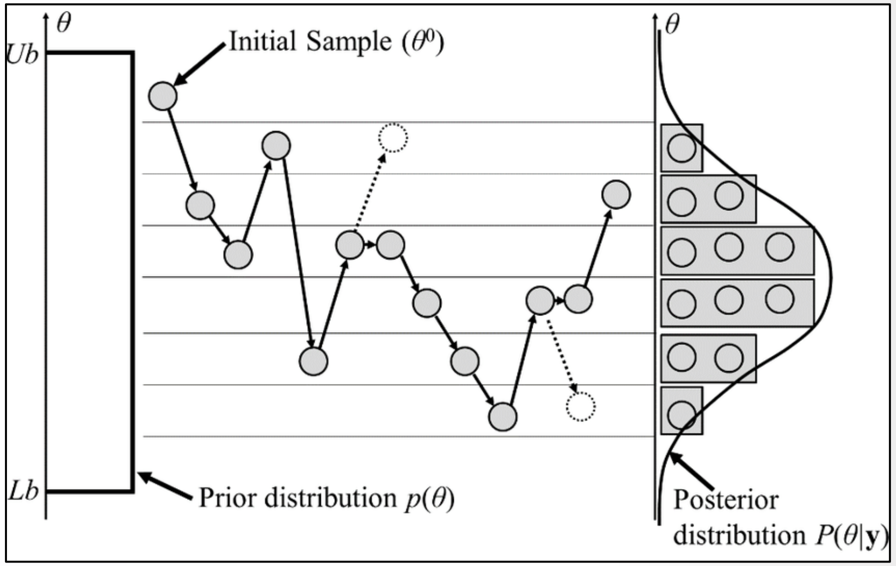
MCMC Backup Slides

What is Markov Chain Monte Carlo (MCMC)?

MCMC is a class of algorithms for sampling from complex probability distributions

It combines two ideas:

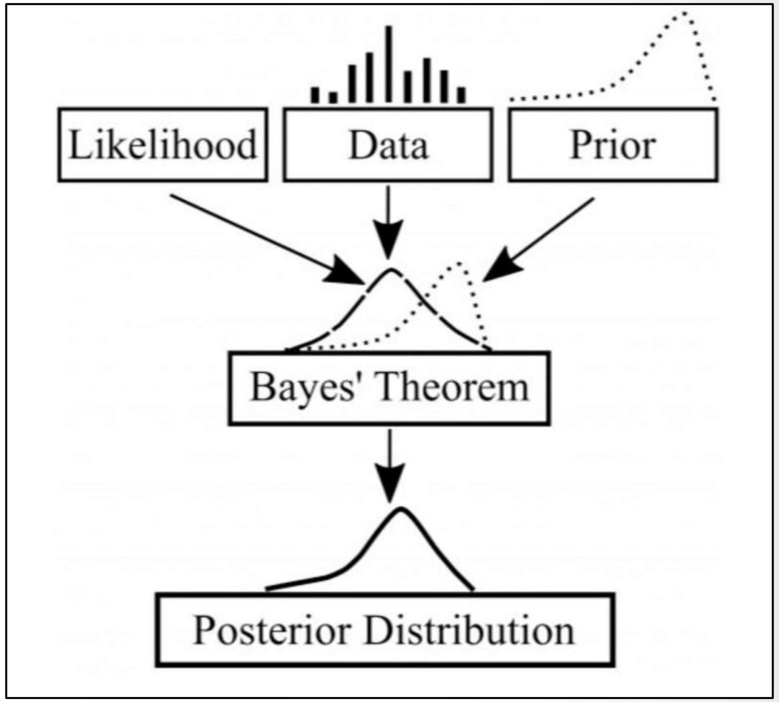
- **Markov Chain**: *Sequence* where each sample depends only on the previous one (memoryless property)
- **Monte Carlo**: Uses *random samples* to estimate properties (e.g., means, integrals) of distributions



What is Markov Chain Monte Carlo (MCMC)?

- **Bayes theorem in practice**: Given $\mathcal{D}$ data and θ parameter values

$$\overbrace{P(\theta|\mathcal{D})}^{\text{posterior}} = \frac{\overbrace{P(\mathcal{D}|\theta)}^{\text{likelihood}}\ \overbrace{P(\theta)}^{\text{prior}}}{\underbrace{P(\mathcal{D})}_{\text{Evidence}}}$$



- MCMC estimates $P(\theta|\mathcal{D})$ without finding $P(\mathcal{D})$ through

$$P(\theta|\mathcal{D}) \propto P(\mathcal{D}|\theta)P(\theta)$$

MCMC in practice

- **Prior information $P(\theta)$:**

```
def log_prior(theta):
    H0, Omega0, M = theta
    if 50 < H0 < 100 and 0.0 < Omega0 < 0.5 and -50.0 < M < 50.0:
        return 0.0
    return -np.inf
```

- **Prior information $P(\mathcal{D}|\theta)$:**

```
def log_probability(theta):
    lp = log_prior(theta)
    if not np.isfinite(lp) or np.isnan(lp):
        return -np.inf
    lp += CC_Cov().loglkl(theta) + Likelihood_Pantheon_SH0ES().loglkl(theta)
    return lp
```

MCMC in practice

- **Boundary conditions:**

```
# MCMC part initialization
nwalkers = 50
initial = np.array([70,0.3,-19])
ndim = len(initial)
p0 = [np.array(initial) + 1e-5 * np.random.randn(ndim) for i in range(nwalkers)]
```

- **Running MCMC:**

```
#MCMC setup
sampler = emcee.EnsembleSampler(nwalkers, ndim, log_probability, pool=pool)
```

```
sampler.run_mcmc(p0, 5000, progress=True)
```

ChainConsumer Plotting

- **Preparing the MCMC output:**

```
# Save output without flattening
flat_samples = sampler.get_chain(flat=True)
log_prob = sampler.get_log_prob(flat=True)
results = np.c_[flat_samples, log_prob]
np.savetxt(filename, results, delimiter=' ')
```

- **Plotting using ChainConsumer:**

```
# Print corner plot
params = ['H0', 'Om', 'M']
chain = Chain.from_emcee(sampler, params, "an emcee chain")
consumer = ChainConsumer().add_chain(chain)
fig = consumer.plotter.plot()
fig.savefig('First_results.pdf')
```