Monte Python: all run and analyze options

Benjamin Audren

École Polytechnique Fédérale de Lausanne

29/10/2014

Outline

Outline

Methods

```
python montepython/MontePython.py run -h
```

Choosing the Sampling Method

- **flag:** `-m [MH, NS, CH, IS, Der]`

Cholesky Decomposition

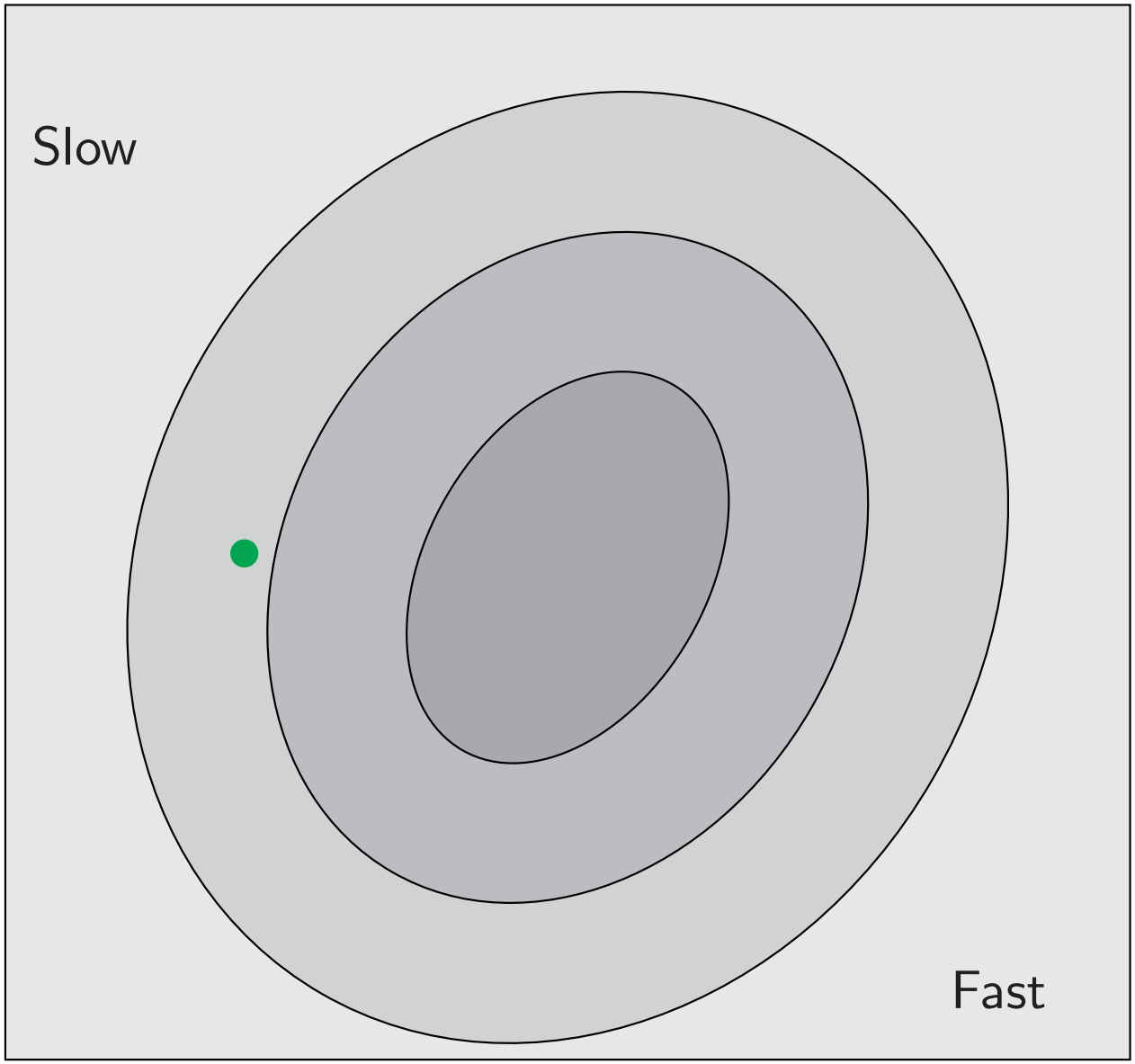
Idea

Cosmological parameters are slow to update (CLASS), but **nuisance** parameters can be fast.
If **varied together**, this distinction is lost.
But, there are **correlations between them!**

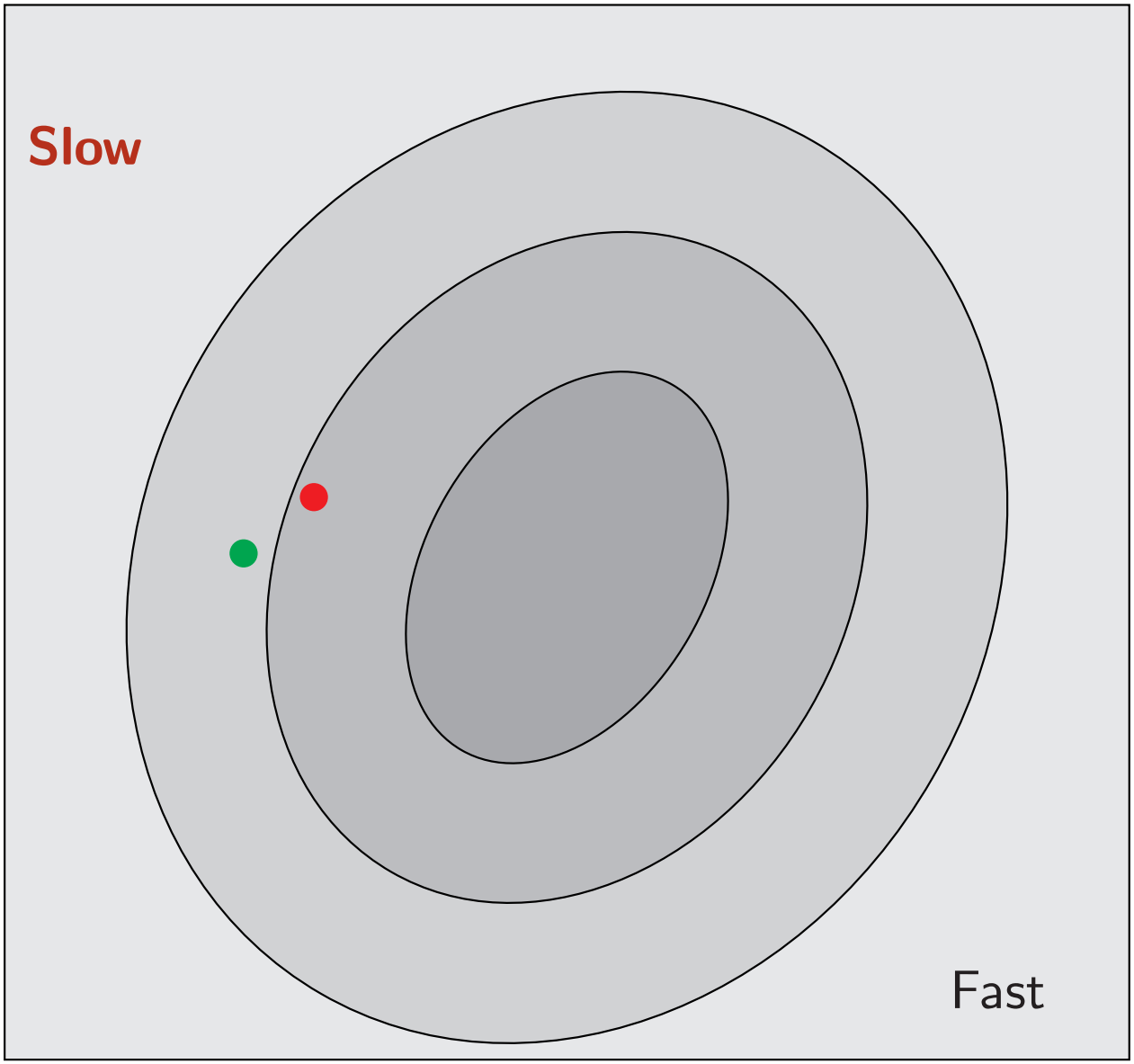
Proposition

Instead of varying all the parameters at each step,
we vary **both fast and slow** some of the time, and
only the fast one the rest of the time.

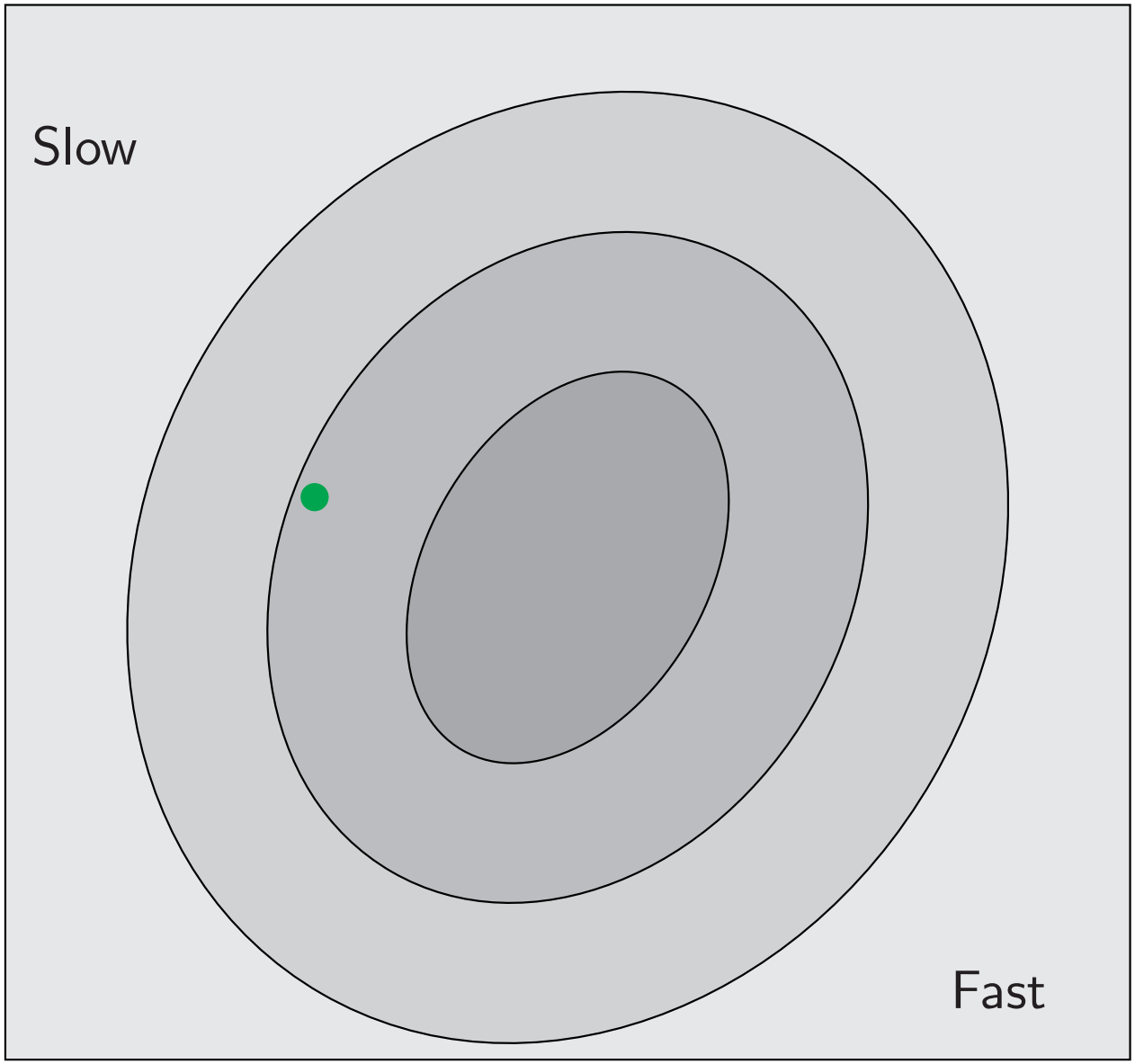
Cholesky Decomposition and Over-sampling



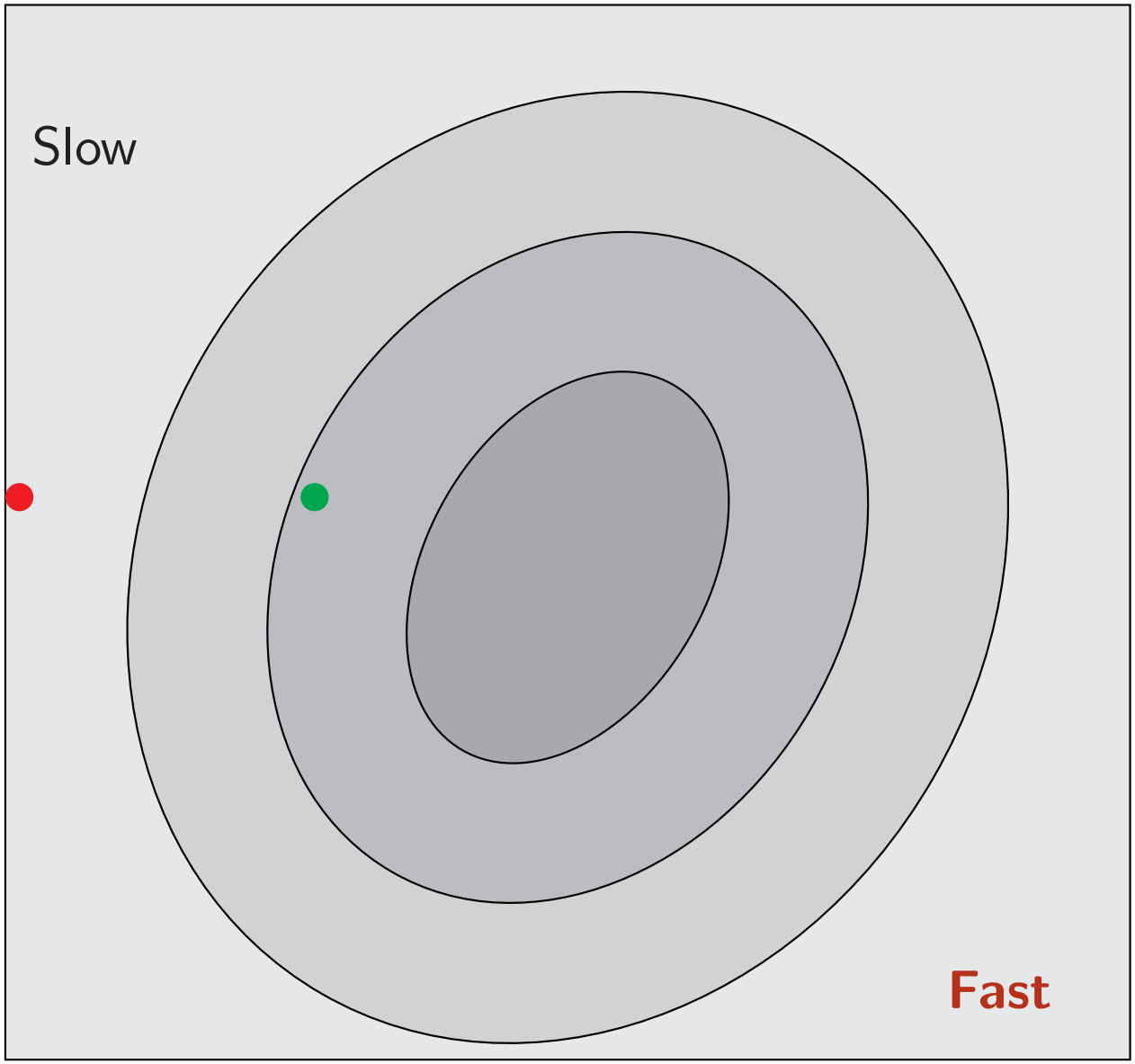
Cholesky Decomposition and Over-sampling



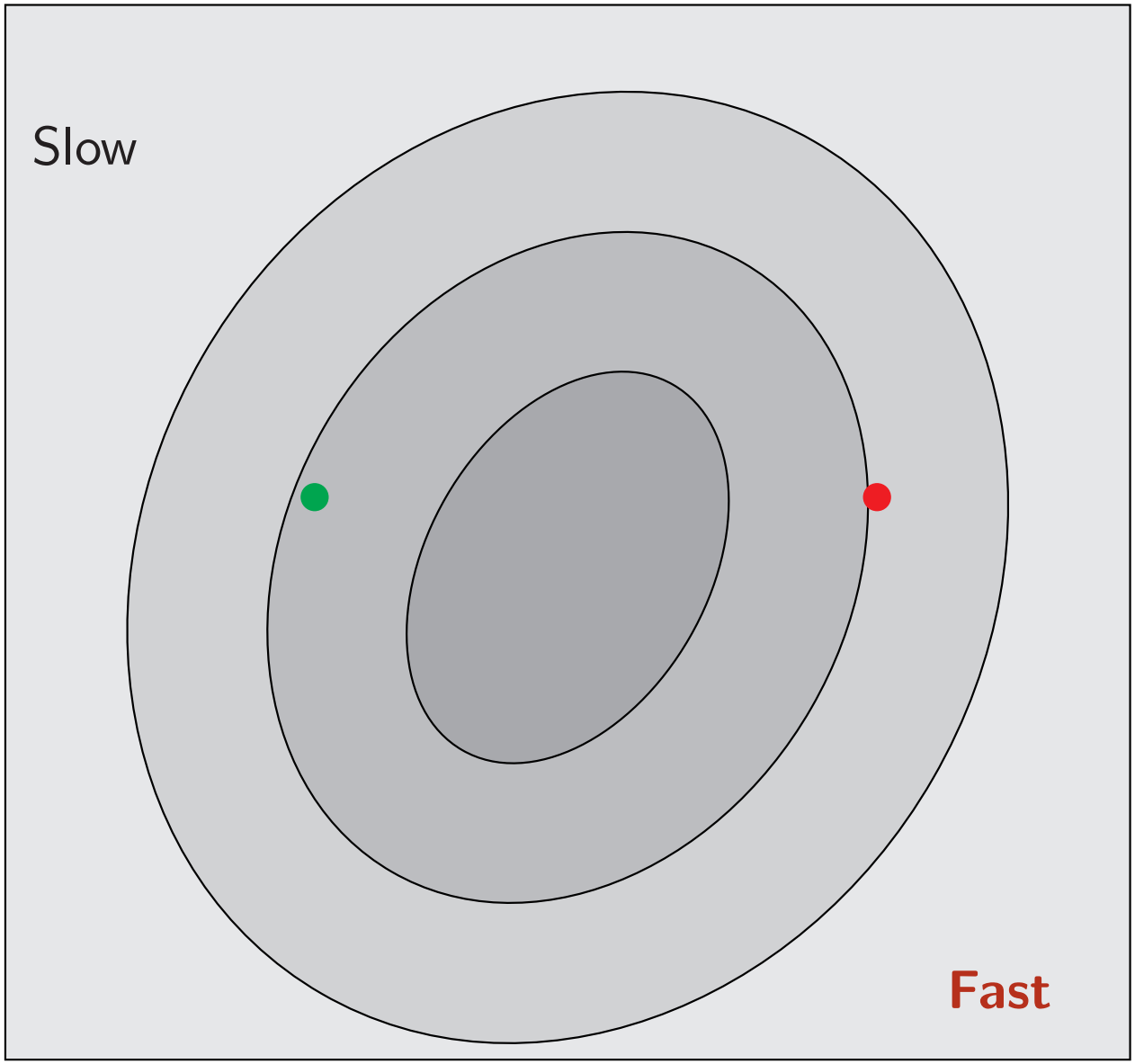
Cholesky Decomposition and Over-sampling



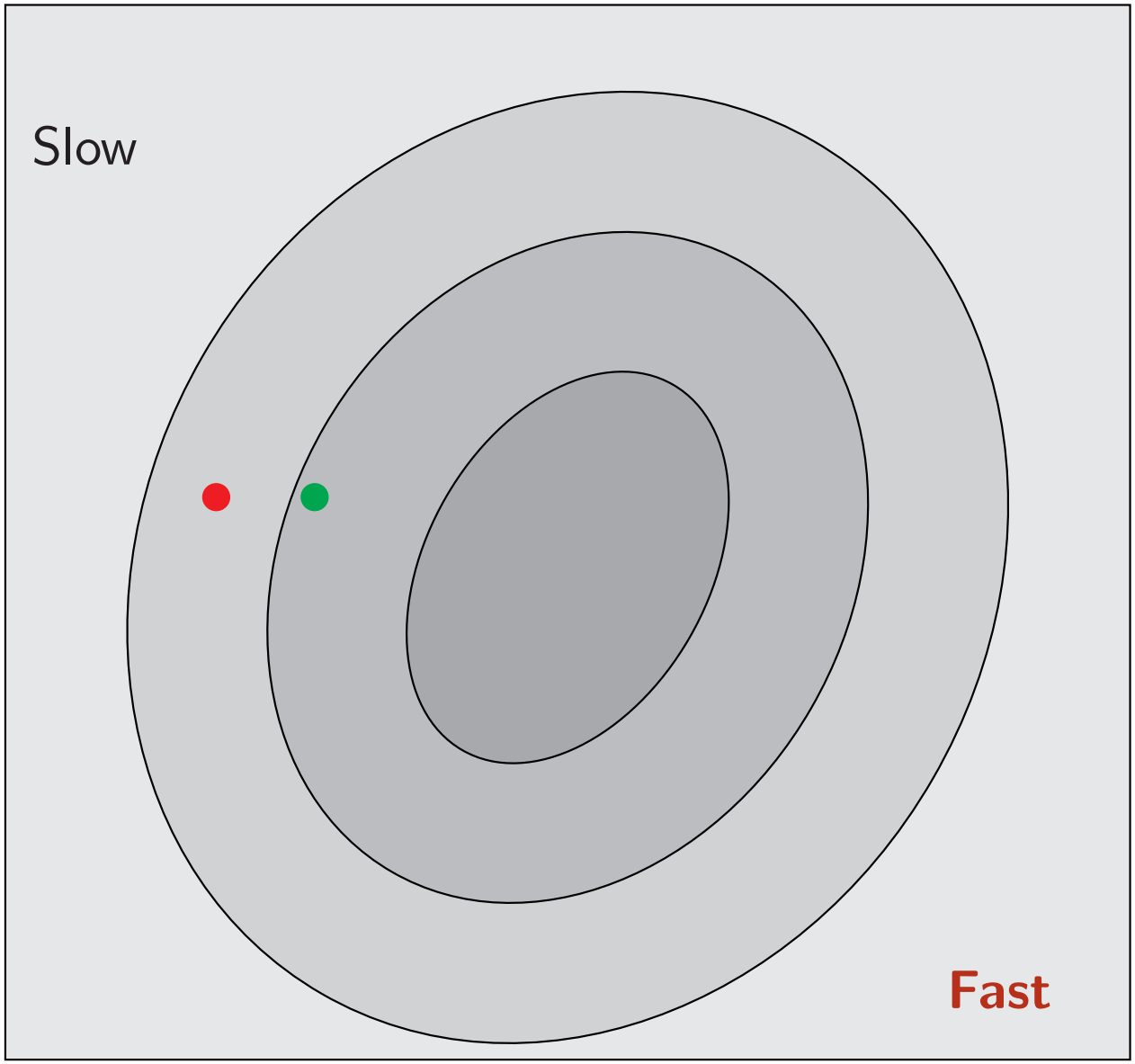
Cholesky Decomposition and Over-sampling



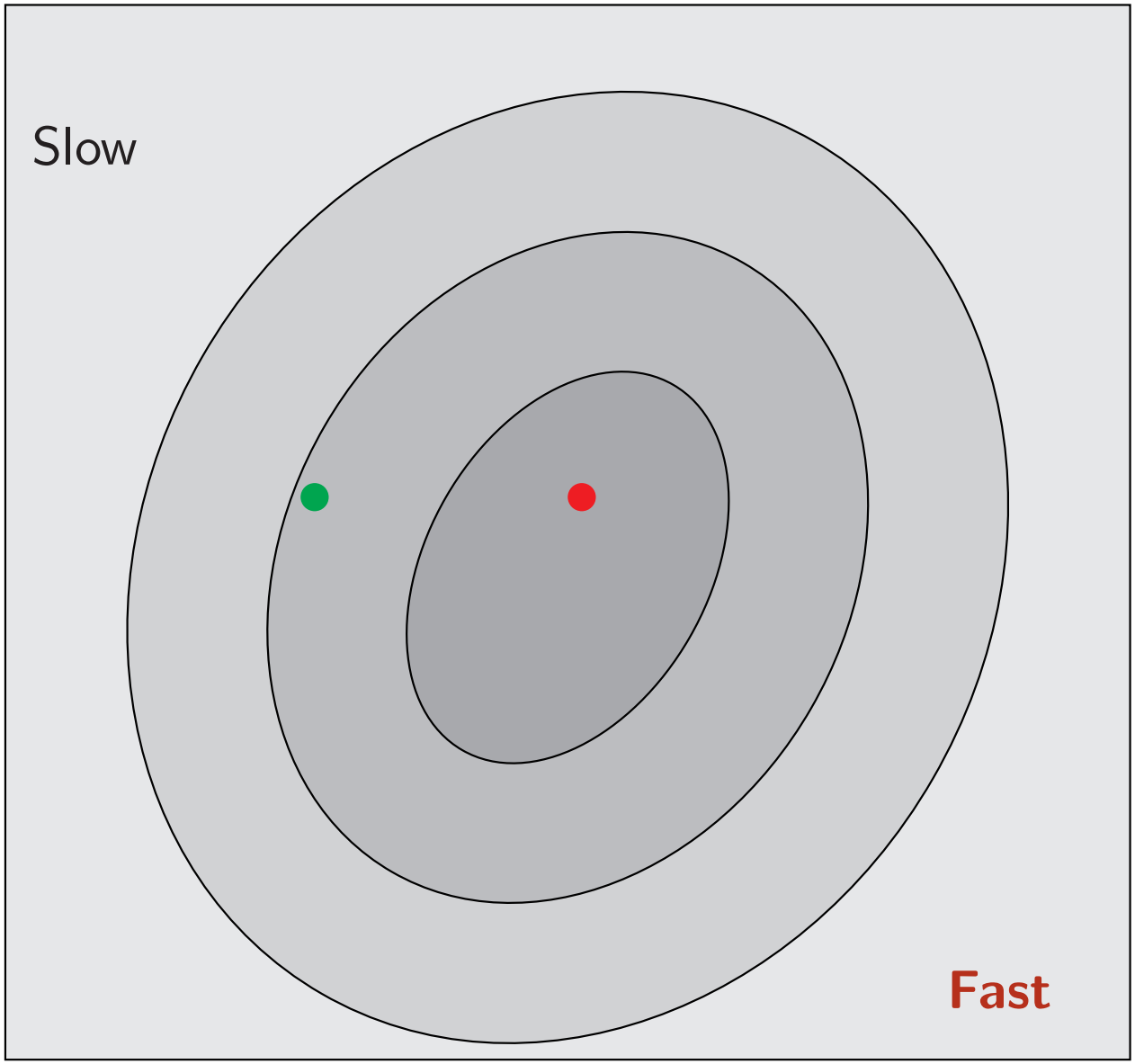
Cholesky Decomposition and Over-sampling



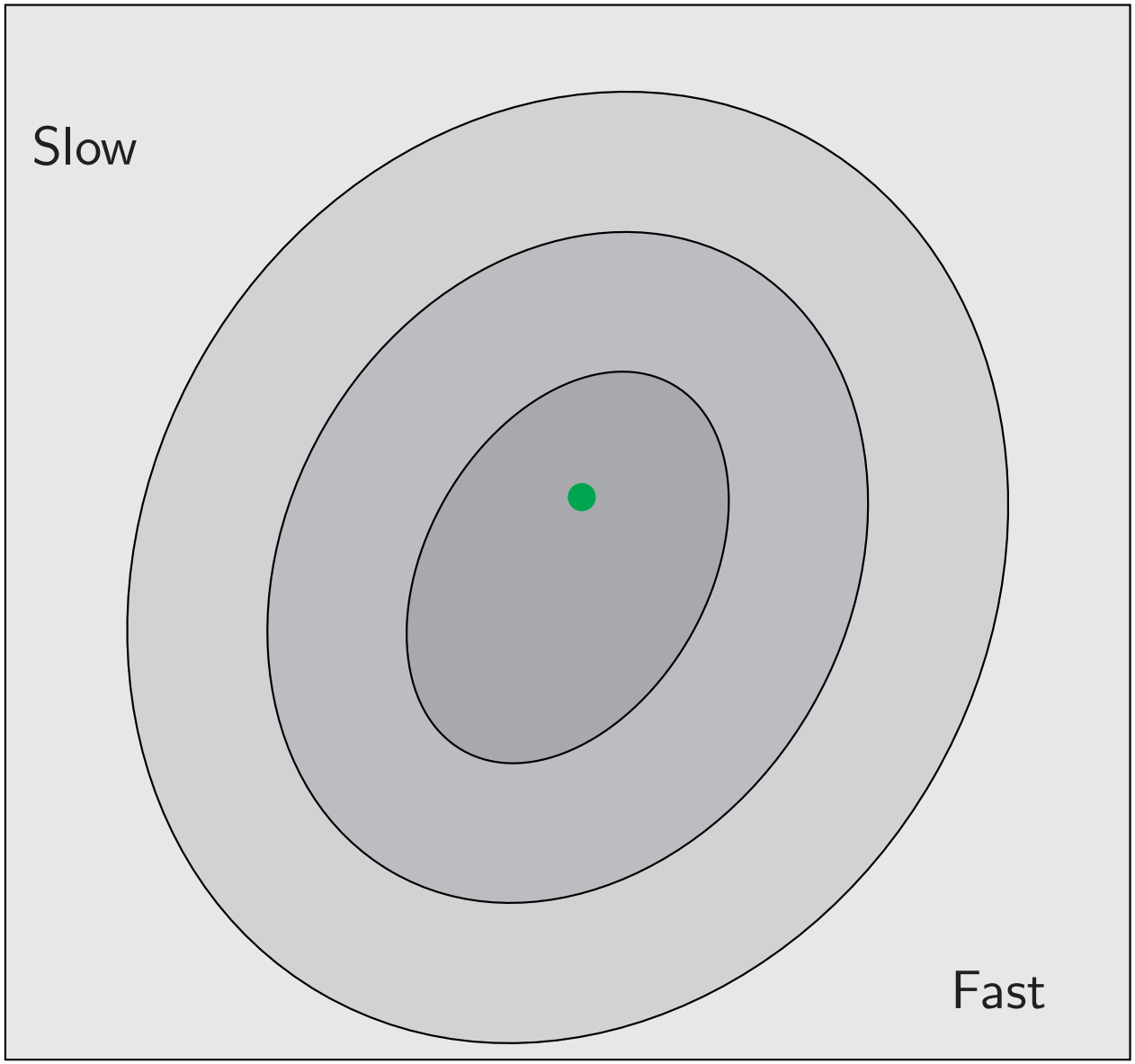
Cholesky Decomposition and Over-sampling



Cholesky Decomposition and Over-sampling



Cholesky Decomposition and Over-sampling



Metropolis Hastings

Basic Options

- **Number of steps**: `-N 1000`
- **Jumping factor**: `-f 1.4`
- **Jumping strategy**: `-j [global, sequential, fast]`
- **Best-fit file**: `-b`, `--bestfit` best-fit point
- **Covmat file**: `-c`, `--covmat` proposal density
- Restart: `-r filename` restart from chain?

Metropolis Hastings

Tricks

Computing the likelihood in one point

```
python montepython/MontePython.py run -o chains/jla -N 1 -f 0
```

Cholesky decomposition

```
python montepython/MontePython.py run -o chains/jla -j fast
```

Nested Sampling

Full description at `http://monte-python.readthedocs.org/en/latest/nested.html`

Principle

- All options from MultiNest can be used
- Simply use `--NS_option_from_MultiNest`
- Most important one: `--NS_n_live_points 100`

Strategy

- separate subfolder (NS)
- chains are translated with `info chains/folder/NS`

Warning

- To see the options, either edit `montepython/nested_sampling.py`
- or `pip install pymultinest (--user)` then `run --help`

Cosmo Hammer

Full description at http://www.astro.ethz.ch/refregier/research/Software/cosmohammer/

Principle

- All options from CosmoHammer can be used
- Simply use `--CH_option_from_CosmoHammer`
- Most important one: `--CH_walkersRatio`

Strategy

- separate subfolder (CH)
- chains are translated with `info chains/folder/CH`

Doc

- To see the options, either edit `montepython/cosmo_hammer.py`

Importance Sampling

Syntax

```
run -m IS -o chains/new -p new.param \
          --IS_starting_folder chains/old
```

Warning

- `new.param` and `old.param` should differ **only through the list of experiments**, and by a mildly constraining likelihood
- Notably, the updated list of experiment can not contain **more nuisance parameters**.

Adding a derived parameter

Syntax

```
run -o existing_folder -m Der \
    --Der-target-folder non_existing_folder \
    --Der-param-list Omega_Lambda
```

Warning

You obviously can add only derived parameters known to CLASS

Outline

Reminder

```
python montepython/MontePython.py info --help
```

Options when analyzing

Main options

- compare several folders `info folder_1 folder_2`
- Remove the mean-likelihood with `--no-mean`
- Only compute the covmat with `--noplot`
- Output all sub plots with `--all`

Using an extra file for plotting options

Alternative to command line

- example in `plot_files/example.plot`
- syntax: `--extra plot_files/example.plot`

Using an extra file for plotting options

Alternative to command line

- example in `plot_files/example.plot`
- syntax: `--extra plot_files/example.plot`

4 exclusive arguments

- **redefine**: dict for **combining parameters**
- **to change**: dict for **renaming**
- **new scales**: dict for **rescaling**
- **to plot**: list for **plotting new names**

Renaming parameters

```
info.to_change  = {'Omega_cdm': r'$\Omega_{\rm cdm}$', 'alpha': '$\gamma$'}
info.new_scales = {'$\gamma$': 10}
info.to_plot    = [r'$\Omega_{\rm cdm}$', '$\gamma$', 'beta']
```

Full example

```
import matplotlib.pyplot as plt

info.redefine = {'log_alpha': 'log_alpha/math.log(10)',
                 'log_sound_speed_sq': 'log_sound_speed_sq/math.log(10)'}
info.to_change = {'log_alpha': r'$\log_{10}(\alpha)$',
                  'log_sound_speed_sq': r'$\log_{10}(c_\chi^2)$',
                  'Y_dm': 'Y'}

info.to_plot    = [r'$\log_{10}(\alpha)$', r'$\log_{10}(c_\chi^2)$', 'Y']

info.fontsize = 24
info.ticksize = 12
info.ticknumber = 5
info.decimal = 2
info.legend-style = 'top'

info.bins = 16

info.cm = [(0.99843, 0.25392, 0.14765, 1.),]
info.cmaps = [plt.cm.Reds]
```

Beware of non-raw strings

Python is picky about that