From a decay law to a likelihood

INITIAL-VALUE PROBLEM

$$y'(t) = -ky(t), \quad y(0) = 1$$

COMPLETE EXAMPLE

A rate from three measurements

```
%%%
Eleven solution times, three measurements.
%%%
flatppl_compat = "0.1"
ode = load_module(
    "https://raw.githubusercontent.com/
    flatppl-community/flatppl-numerics-ode/
    a56b4977b3ca4fced2e986bcbbdb6930b9d8c04a/
    main.flatppl"
)
t_grid = linspace(0.0, 1.0, 11)
y_data = [1.02, 0.57, 0.38]
rate = elementof(posreals)
rhs(t, y, k) = -k .* y
states = ode.vern7(rhs, [1.0], t_grid, rate)
Y = rowstack(states)
mu = [Y[1, 1], Y[6, 1], Y[11, 1]]
reading ~ Normal.(mu, 0.05)
K = kernelof(reading, rate = rate)
L = likelihoodof(K, y_data)
score = logdensityof(L, record(rate = 1.0))
predict = functionof(mu, rate = rate)
fitted = predict(rate = 1.0)
```

Supply a complete time grid

Use the fixed-step solver from `flatppl-numerics-ode`. `linspace(0.0, 1.0, 11)` includes both endpoints. Each consecutive pair defines one time step.

Use the closed form as a check

`exact = exp.(-1.0 .* t_grid)`
For this derivative, $y(t) = \exp(-kt)$. At $k = 1$, compare the numerical and exact solutions at each grid point before attaching the measurement law.

Inspect the residuals

`resid = (y_data .- fitted) ./ 0.05`
In units of measurement noise, the residuals at rate one are approximately 0.40, −0.73, and 0.24.

ANNOTATED LINES

A Derivative and initial condition — 12–13

```
rhs(t, y, k) = -k .* y
states = ode.vern7(rhs, [1.0], t_grid, rate)
```

Pass time, the state vector and the rate to `rhs`. Starting from `[1.0]`, `vern7` integrates over the fixed grid. Passing `rate` explicitly makes the trajectory a function of the inferred parameter.

B Select the measured times — 15–16

```
Y = rowstack(states)
mu = [Y[1, 1], Y[6, 1], Y[11, 1]]
```

Stack one time per row and one state component per column with `rowstack`. With one-based indexing, rows 1, 6, and 11 give times 0, 0.5, and 1.

C Gaussian measurement errors — 18–20

```
reading ~ Normal.(mu, 0.05)
K = kernelof(reading, rate = rate)
L = likelihoodof(K, y_data)
```

Use the three selected states as Gaussian means with common standard deviation 0.05. Declare the positive rate as the kernel input and condition on the three readings.

D Likelihood at unit rate — 22–24

```
score = logdensityof(L, record(rate = 1.0))
predict = functionof(mu, rate = rate)
fitted = predict(rate = 1.0)
```

At `rate = 1` the three independent Gaussian readings give a log likelihood of 5.85410.

Time	0	0.5	1
Predicted	1.00000	0.60653	0.36788
Observed	1.02	0.57	0.38

Information at the initial time

All rates give the same initial condition. The first reading therefore adds a constant to the log likelihood. The two later readings constrain the decay rate.

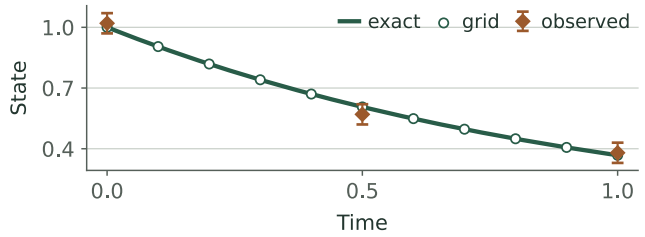
Sensitivity to the rate

The exact sensitivity is $\partial_k y = -t\exp(-kt)$. At $k = 1$ the sensitivities at times 0, 0.5, and 1 are 0, −0.30327, and −0.36788.

For an observed reading z and predicted state y, its score contribution is $(z - y)\frac{\partial_k y}{\sigma^2}$.

TIME GRID AND MEASUREMENT MODEL

A trajectory and three readings



The line is the exact solution at $k = 1$. Circles mark the solver grid. Diamonds and error bars show the three observations and one measurement standard deviation.

Separate solver and data indices

The solver returns eleven states for three measured times. Only those three states enter the likelihood. Additional grid points refine the integration.

Mesh refinement

`fine_t = linspace(0.0, 1.0, 21)`
Compare the two solutions at the same measured times.

Time	0	0.5	1
Row on 11-point grid	1	6	11
Row on 21-point grid	1	11	21

Extend to coupled states

An initial vector with several entries describes a coupled system. Return a derivative vector of the same length. Choose the observed state components from the corresponding solution columns.

Add a prior for the decay rate

```
k ~ LogNormal(0, 0.5)
prior = lawof(record(rate = k))
posterior = bayesupdate(L, prior)
```

The prior uses the same positive rate field as the likelihood.

Jessica Cox · bcox@mpp.mpg.de