Scientific Computing in Python Lecture 4

ODE integration, root finding, and parameter estimation

Zan Ahmad

Winter 2026

Newton's cooling equation

$$\dot{T} = -k(T - T_a), \qquad T(0) = T_0.$$

Newton's cooling equation

$$\dot{T} = -k(T - T_a), \qquad T(0) = T_0.$$

Set $u = T - T_a$. Then $\dot{u} = -ku$, and k has units of inverse time.

Exact solution

$$\frac{\mathrm{d}u}{u} = -k\ \mathrm{d}t \quad \Longrightarrow \quad \log\left|\frac{u(t)}{u(0)}\right| = -kt.$$

Exact solution

$$\frac{\mathrm{d}u}{u} = -k\ \mathrm{d}t \quad \Longrightarrow \quad \log\left|\frac{u(t)}{u(0)}\right| = -kt.$$

$$T(t) = T_a + (T_0 - T_a)e^{-kt}.$$

For $u(0) = 0$, use the constant solution $u(t) = 0$.

Time scales and forward Euler

$$T_{n+1} = T_n - hk(T_n - T_a), \qquad u_{n+1} = (1 - kh)u_n.$$

Time scales and forward Euler

$$T_{n+1} = T_n - hk(T_n - T_a), \qquad u_{n+1} = (1 - kh)u_n.$$

Decay requires $|1 - kh| < 1$, or $0 < kh < 2$.

Monotone versus oscillatory decay

$0 < kh \leq 1$: u_n preserves sign and decays,

Monotone versus oscillatory decay

$$0 < kh \leq 1: \quad u_n \text{ preserves sign and decays,}$$

$$1 < kh < 2: \quad u_n \text{ alternates sign while decaying.}$$

The second behavior is numerically stable but inconsistent with monotone cooling in this model.

Adaptive integration

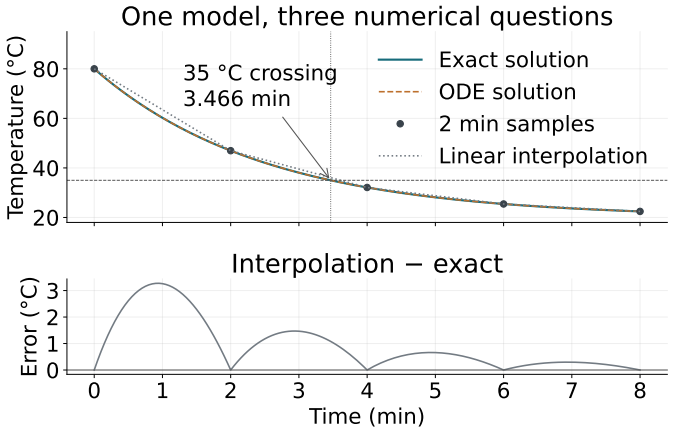
```
sol = solve_ivp(rhs, (0, 8), [80],
                t_eval=times, rtol=1e-8, atol=1e-10)
```

Adaptive integration

```
sol = solve_ivp(rhs, (0, 8), [80],
                t_eval=times, rtol=1e-8, atol=1e-10)
```

Compare $T_{\text{num}}(t_i)$ with $20 + 60e^{-0.4t_i}$.

Cooling trajectory and interpolation error



Error under refinement

$$E(h) = \max_i |T_h(t_i) - T(t_i)|.$$

Error under refinement

$$E(h) = \max_i |T_h(t_i) - T(t_i)|.$$

For first-order Euler convergence, $E(h)/E(h/2)$ approaches 2 when the step is sufficiently small and roundoff does not dominate.

Threshold time as a root

$$T_* - T_a = (T_0 - T_a)e^{-kt_*}.$$

Threshold time as a root

$$T_* - T_a = (T_0 - T_a)e^{-kt_*}.$$

$$t_* = \frac{1}{k} \log \frac{T_0 - T_a}{T_* - T_a}.$$

A threshold calculation

For $T_0 = 80$, $T_a = 20$, $k = 0.4$ and $T_* = 40$,

$$t_* = 2.5 \log 3 \approx 2.74653 \text{ min.}$$

A threshold calculation

For $T_0 = 80$, $T_a = 20$, $k = 0.4$ and $T_* = 40$,

$$t_* = 2.5 \log 3 \approx 2.74653 \text{ min}.$$

The logarithm's argument is dimensionless and greater than one.

Linear interpolation

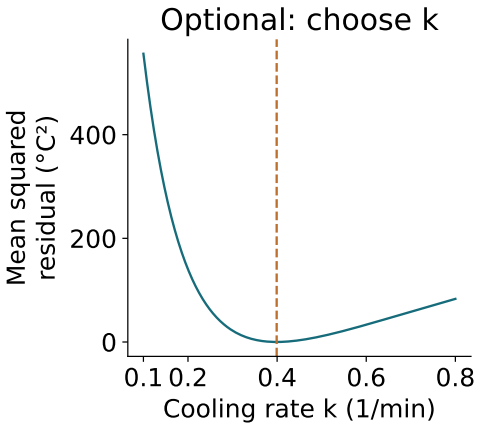
$$\widetilde{T}(t) = (1-\alpha)T_i + \alpha T_{i+1}, \qquad \alpha = \frac{t - t_i}{t_{i+1} - t_i}.$$

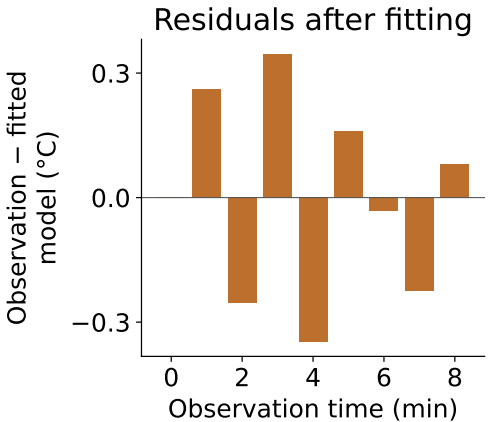
Linear interpolation

$$\widetilde{T}(t) = (1-\alpha)T_i + \alpha T_{i+1}, \qquad \alpha = \frac{t - t_i}{t_{i+1} - t_i}.$$

Here $0 \leq \alpha \leq 1$ only within the sampled interval.

Parameter objective and residuals





Interpolation error for cooling

$$T(t) - \widetilde{T}(t) = \frac{T''(\xi)}{2}(t - t_i)(t - t_{i+1}).$$

Interpolation error for cooling

$$T(t) - \widetilde{T}(t) = \frac{T''(\xi)}{2}(t - t_i)(t - t_{i+1}).$$

Cooling from above ambient has $T'' > 0$. The product is nonpositive inside the interval, so the secant lies above the exact curve.

Fitting the cooling coefficient

$$J(k) = \frac{1}{n} \sum_i [T_a + (T_0 - T_a)e^{-kt_i} - y_i]^2.$$

Fitting the cooling coefficient

$$J(k) = \frac{1}{n}\sum_i [T_a + (T_0 - T_a)e^{-kt_i} - y_i]^2.$$

$$\frac{\partial T(t;k)}{\partial k} = -t(T_0 - T_a)e^{-kt}.$$

Sensitivity in time

At $t = 0$, the temperature is independent of k. At very long times it is nearly ambient and again weakly sensitive.

Sensitivity in time

At $t = 0$, the temperature is independent of k. At very long times it is nearly ambient and again weakly sensitive. For $t > 0$, the magnitude te^{-kt} is largest at $t = 1/k$. Measurements near that scale are informative about k.