Ordinary differential equations

Phys 750 Lecture 7

Ordinary Differential Equations

- Most physical laws are expressed as differential equations
- These come in three flavours:
  - initial-value problems
  - boundary-value problems
  - eigenvalue problems

Ordinary Differential Equations

- Most physical laws are expressed as differential equations
- These come in three flavours:
  - initial-value problems
  - boundary-value problems
  - eigenvalue problems

first order

$$x'(t) = F(x(t), t)$$
$$x(0) = x_0$$

second order

$$x''(t) = F(x(t), x'(t), t)$$
$$x(0) = x_0$$
$$x'(0) = v_0$$

Ordinary Differential Equations

- Most physical laws are expressed as differential equations
- These come in three flavours:
  - initial-value problems
  - boundary-value problems
  - eigenvalue problems

$$\forall\, x \in \mathcal{R}:$$

$$u''(x) = F(u(x), u'(x); x)$$

$$\forall\, x \in \partial\mathcal{R}:$$

$$u(x) = \alpha(x) \text{ or}$$

$$u'(x) = \beta(x)$$

Ordinary Differential Equations

- Most physical laws are expressed as differential equations
- These come in three flavours:
  - initial-value problems
  - boundary-value problems
  - eigenvalue problems

$$\forall\, x \in \mathcal{R} :$$

$$u''(x) = F(u(x), u'(x); x; \lambda)$$

$$\forall\, x \in \partial\mathcal{R} :$$

$$u(x) = \alpha(x) \text{ or}$$
$$u'(x) = \beta(x)$$

solutions only at specific eigenvalues

Ordinary Differential Equations

- In principle, the initial value problem ODE can be forward integrated from its specified starting point

$$\frac{dx}{dt} = F(x(t), t) \qquad x(0) = x_0 \longrightarrow x(t) = x_0 + \int_0^t dt'\, F(x(t'), t')$$

- Need to generate a numerical estimate of the integral on the rhs of the formal solution

Ordinary Differential Equations

- As usual, we chop the real time variable into discrete time steps $\Delta t = t_{i+1} - t_i$

$$x(t + \Delta t) = x(t) + \int_t^{t+\Delta t} dt' \, F(x(t'), t')$$

$$x(t_{i+1}) = x(t_i) + \int_{t_i}^{t_{i+1}} dt' \, F(x(t'), t')$$

- If Δt is sufficiently small, then the integral is well-approximated by a low-order estimate of the area

Ordinary Differential Equations

- Various first-order approximations:

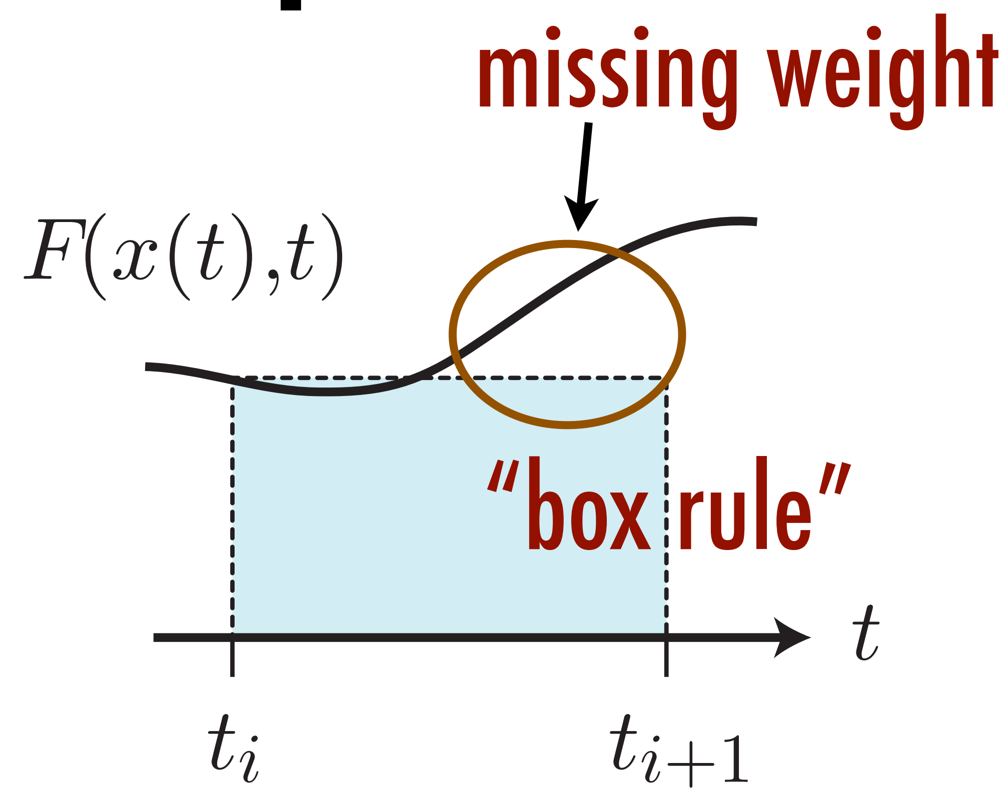
$$x_{i+1} = x_i + F_i \Delta t$$

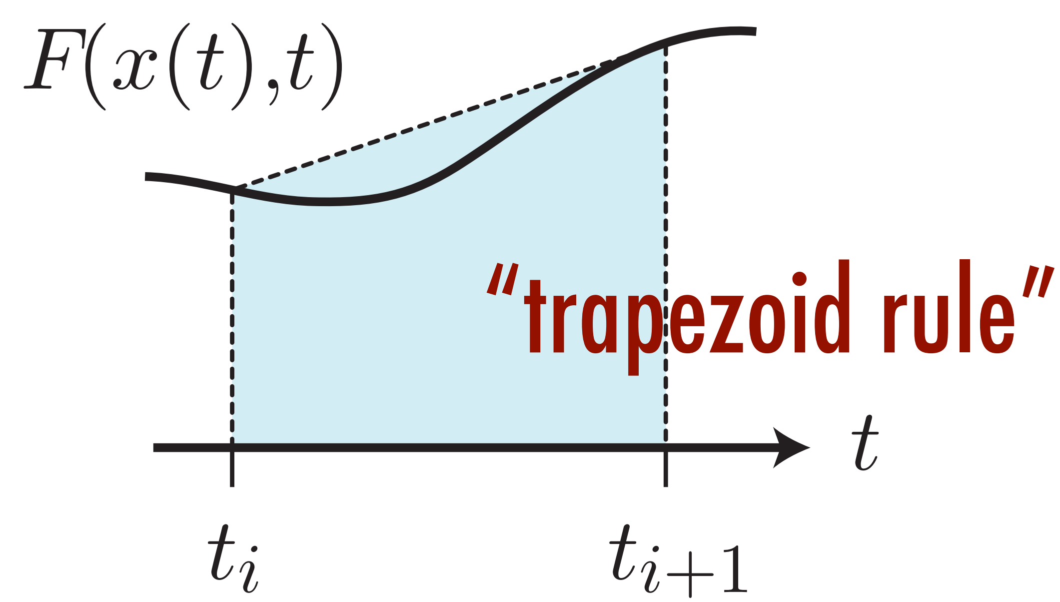
$$x_{i+1} = x_i + \frac{1}{2}\left(F_i + F_{i+1}\right)\Delta t$$

where in the future

$$x_i = x(t_i), F_i = F(x(t_i), t_i)$$

- The choices of box and trapezoid integration correspond to the so-called Euler and Picard methods





Ordinary Differential Equations

- Algorithm for the Euler method is very simple
- Accuracy of the method is low, and large errors accumulate over time
- Not necessarily energy-conserving

- Set x_0 to its initial value
- Step through each t_i $(i \geq 0)$:
  - compute $F_i = F(x_i, t_i)$
  - $x_{i+1} = x_i + F_i \Delta t$
  - $t_{i+1} = t_i + \Delta t$

Ordinary Differential Equations

- Picard method requires a self-consistent solution
- Accurate but slow
- May not converge for too large a choice of time step

- Set x_0 to its initial value
- Step through each t_i $(i \geq 0)$:
  - compute $F_i = F(x_i, t_i)$
  - $t_{i+1} = t_i + \Delta t$
  - compute $x_{i+1}^{(1)}$ via Euler
  - Loop over $k = 1, 2, 3, \ldots$
    - compute $F_{i+1}^{(k)} = F(x_{i+1}^{(k)}, t_{i+1})$
    - $x_{i+1}^{(k+1)} = x_i + \frac{1}{2}\left(F_i + F_{i+1}^{(k)}\right)\Delta t$
    - Exit loop if $|x_{i+1}^{(k+1)} - x_{i+1}^{(k)}| < \epsilon$
  - $x_{i+1} = x_{i+1}^{(k_{\max})}$

Ordinary Differential Equations

- Systematic expansion: replace dummy variable by and Taylor expand the integrand $t' = t + \delta t$

$$F(x(t+\delta t), t+\delta t) = F(x(t),t) + \frac{\partial F}{\partial x} x'(t)\delta t + \frac{\partial F}{\partial t}\delta t + O(\delta t)^2$$

- Integration over $0 < \delta t < \Delta t$ yields

not necessarily available to us

$$x_{i+1} = x_i + F_i \Delta t + \frac{1}{2}\left[F_i \left.\frac{\partial F}{\partial x}\right|_i + \left.\frac{\partial F}{\partial t}\right|_i \right] (\Delta t)^2 + \cdots$$

- Truncation at first order corresponds to the Euler method

Ordinary Differential Equations

- According to the mean value theorem, an exact truncation is of the form
$$x_{i+1} = x_i + F(x(\tau), \tau)\Delta t,\ \tau \in [t_i, t_{i+1}]$$
- F is evaluated at some intermediate point
- Ideal value of τ absorbs all curvature corrections
- Possibility of systematic improvements

Ordinary Differential Equations

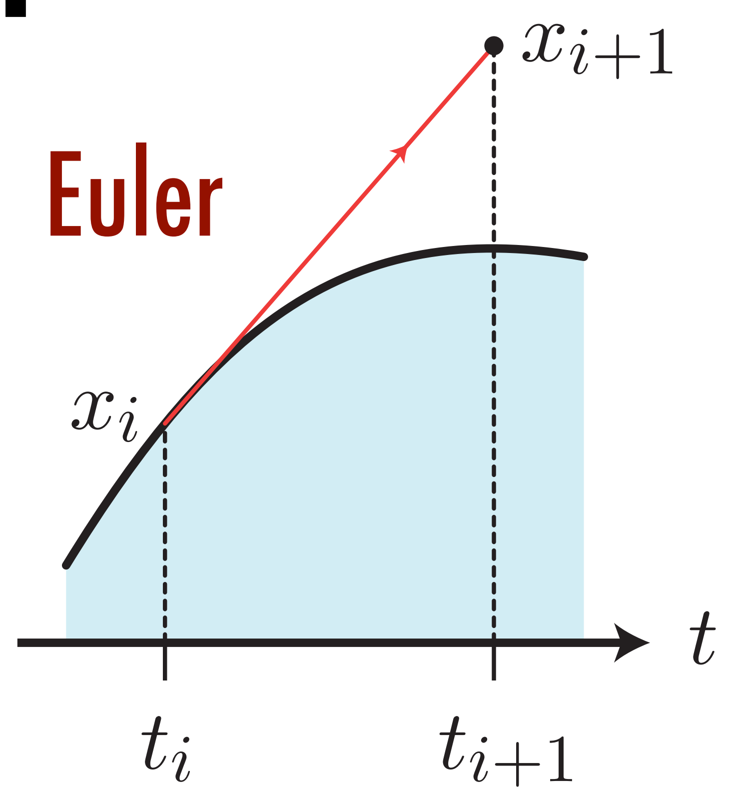
- E.g., second-order Runge-Kutta

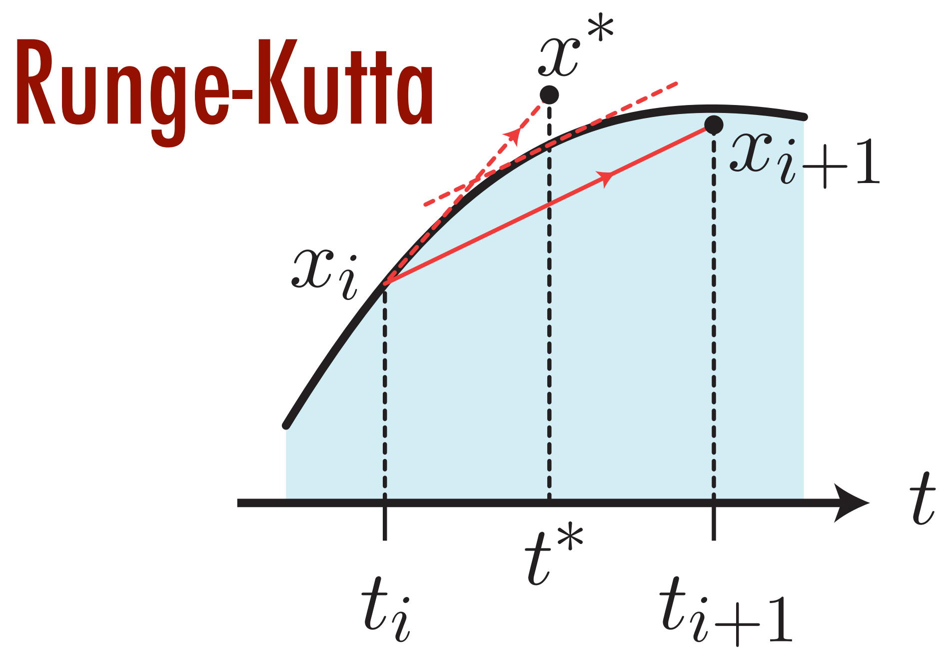
$$x_{i+1} = x_i + F(x^*, t^*)\Delta t$$

$$x^* = x_i + \frac{1}{2}F(x_i, t_i)\Delta t$$

$$t^* = t_i + \frac{1}{2}\Delta t$$

- local errors at $\mathrm{O}(\Delta t)^2$





Runge-Kutta Schemes

- Exact evolution over a small time step:

$$x(t+\Delta t) = x(t) + \int_0^{\Delta t} d(\delta t)\, F(x(t+\delta t), t+\delta t)$$

- Expand both sides in a small time increment:

$$\begin{aligned} x(t+\Delta t) &= x(t) + x'(t)\Delta t + \frac{1}{2}x''(t)(\Delta t)^2 + \frac{1}{6}x'''(t) + \cdots \\ &= x(t) + F\Delta t + \frac{1}{2}\big(F_t + F F_x\big)(\Delta t)^2 \\ &\quad + \frac{1}{6}\big(F_{tt} + 2FF_{tx} + F^2 F_{xx} + FF_x^2 + F_t F_x\big)(\Delta t)^3 + \cdots \end{aligned}$$

partial derivatives

Runge-Kutta Schemes

- Runge-Kutta ansatz at order m:

$$x(t + \Delta t) = x(t) + \alpha_1 c_1 + \alpha_2 c_2 + \cdots + \alpha_m c_m$$

- Function evaluation at many points in the interval

$$\begin{aligned}
c_1 &= (\Delta t) F(x, t) \\
c_2 &= (\Delta t) F(x + \nu_{21} c_1, t + \nu_{21} \Delta t) \\
c_3 &= (\Delta t) F(x + \nu_{31} c_1 + \nu_{32} c_2, t + (\nu_{31} + \nu_{32}) \Delta t) \\
c_4 &= (\Delta t) F(x + \nu_{41} c_1 + \nu_{42} c_2 + \nu_{43} c_3, t + (\nu_{41} + \nu_{42} + \nu_{43}) \Delta t) \\
&\vdots
\end{aligned}$$

- m equations and $m + m(m-1)/2$ unknowns $\{\alpha_i, \nu_{ij}\}$

Second-order ODEs

- We have discussed the Euler, Picard, and Runge-Kutta schemes for integrating the first-order initial value problem:

$$x'(t) = F(x(t), t)$$
$$x(0) = x_0$$

- Similar considerations can be applied to the second-order problem:

$$x''(t) = F(x(t), x'(t), t)$$
$$x(0) = x_0$$
$$x'(0) = v_0$$

Second-order ODEs

- Convenient to reinterpret the second-order system as two coupled first-order equations

$$
\begin{aligned}
x''(t) &= F(x(t), x'(t), t) \\
x(0) &= x_0 \\
x'(0) &= v_0
\end{aligned}
\quad \longrightarrow \quad
\begin{aligned}
v'(t) &= A(x(t), v(t), t) \\
x'(t) &= v(t) \\
x(0) &= x_0 \\
v(0) &= v_0
\end{aligned}
$$

- Obvious connection to classical mechanics: velocity v and acceleration model A

Second-order ODEs

- Naive generalization of Euler method to the pair of first order equations
- Some ambiguity in the labelling of time steps

- Set x_0 and v_0 to their initial values
- Step through each t_i ($i \geq 0$):
  - compute $a_i = A(x_i, v_i, t_i)$
  - $v_{i+1} = v_i + a_i \Delta t$
  - $x_{i+1} = x_i + v_i \Delta t$
  - $t_{i+1} = t_i + \Delta t$

Could equally read v_{i+1} and still be correct to $O(\Delta t)$ (Euler-Cromer)

Second-order ODEs

- Can achieve higher order algorithms systematically at the cost of having more time steps involved in each update

$$x_{i+1} = x_i + v_i \Delta t + \frac{1}{2} a_i (\Delta t)^2 + O(\Delta t)^3$$

$$x_{i-1} = x_i - v_i \Delta t + \frac{1}{2} a_i (\Delta t)^2 + O(\Delta t)^3$$

- Adding and subtracting the forward and reverse forms

$$x_{i+1} = 2x_i - x_{i-1} + a_i (\Delta t)^2$$

$$v_i = \frac{x_{i+1} - x_{i-1}}{\Delta t}$$

(Verlet method)

Second-order ODEs

- Verlet method is more numerically stable than Euler
- It is not self-starting: it needs both (x_0, v_0) and (x_{-1}, v_{-1})
- Accuracy can be arbitrarily improved in this way at the cost of more starting points: (x_{-2}, v_{-2}), etc.
- Fortunately, there is an update rule mathematically equivalent to Verlet that is self-starting:

$$x_{i+1} = x_i + v_i \Delta t + \frac{1}{2} a_i (\Delta t)^2$$

$$v_{i+1} = v_i + \frac{1}{2} (a_{i+1} + a_i) \Delta t$$

depends on x_{i+1} only

(self-starting Verlet)

Second-order ODEs

- Runge-Kutta has the advantage of begin self-starting
- 4th order Runge-Kutta for Newton's equations of motion:

$$
\begin{aligned}
k_{1v} &= A(x_i, v_i, t_i)\Delta t \\
k_{1x} &= v_i \Delta t \\
k_{2v} &= A(x_i + \tfrac{1}{2}k_{1x}, v_i + \tfrac{1}{2}k_{1v}, t_i + \tfrac{1}{2}\Delta t) \\
k_{3v} &= A(x_i + \tfrac{1}{2}k_{2x}, v_i + \tfrac{1}{2}k_{2v}, t_i + \tfrac{1}{2}\Delta t) \\
k_{3x} &= \left(v_i + \tfrac{1}{2}k_{2v}\right)\Delta t \\
k_{4v} &= A(x_i + k_{3x}, v_i + k_{3v}, t_i + \Delta t) \\
k_{4x} &= \left(v_i + k_{3x}\right)\Delta t \\
v_{i+1} &= v_i + \tfrac{1}{6}\left(k_{1v} + 2k_{2v} + 2k_{3v} + k_{4v}\right) \\
x_{i+1} &= x_i + \tfrac{1}{6}\left(k_{1x} + 2k_{2x} + 2k_{3x} + k_{4x}\right)
\end{aligned}
$$