PHY604 Lecture 7

September 16, 2025

Today's lecture

- Ordinary differential equations:
  - Euler method
  - Runge-Kutta methods and adaptive RK
  - Beyond Runge-Kutta
    - Leapfrog/Verlet/modified midpoint
    - Bulirsch-Stoer Method

Differential equations (Newman Ch. 8)

- One of the major applications of computation to science and engineering is solving differential equations
  - Even for very simple-looking equations if they are "nonlinear," they are difficult or imposible to solve analytically
- Classifications:
  - Initial value problems
  - Boundary value problems
  - Eigenvalue problems
- Often problems are described by systems of coupled differential equations
- As with the other topics, there are many different methods
  - We just want to see the basic ideas and popular methods

Example of system of differential equations: Equations of motion

- We know that the equations of motion for a point particle with mass are given by:

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(t), \qquad \frac{d\mathbf{v}}{dt} = \mathbf{a}(\mathbf{x}, \mathbf{v}, t)$$

- In order to fully describe the trajectory of this particle, we need to specify initial conditions, i.e., the position and velocity, of the particle at the initial time *t* = 0:

$$\mathbf{x}(0) = \mathbf{x}_0, \qquad \mathbf{v}(0) = \mathbf{v}_0$$

Approximating the Equations of Motion

- If we consider a time interval that is sufficiently short, we can approximate the differential by

$$dt \simeq \Delta t$$

- We can then approximate the time derivative of the position by:

$$\frac{d\mathbf{x}}{dt} \simeq \frac{\mathbf{x}(t+\Delta t) - \mathbf{x}(t)}{\Delta t}$$

- Similarly, the time derivative of the velocity can be approximated by

$$\frac{d\mathbf{v}}{dt} \simeq \frac{\mathbf{v}(t+\Delta t) - \mathbf{v}(t)}{\Delta t}$$

Euler's method for integrating the equations of motion

- We can then substitute the approximate derivatives into the equations of motion to obtain:

$$\frac{\mathbf{x}(t+\Delta t)-\mathbf{x}(t)}{\Delta t} \simeq \mathbf{v}(t), \quad \frac{\mathbf{v}(t+\Delta t)-\mathbf{v}(t)}{\Delta t} \simeq \mathbf{a}(\mathbf{x},\mathbf{v},t)$$

- We can then solve for the new values of the position and velocity

$$\mathbf{v}(t+\Delta t) \simeq \mathbf{v}(t) + \mathbf{a}(\mathbf{x},\mathbf{v},t)\Delta t$$

$$\mathbf{x}(t+\Delta t) \simeq \mathbf{x}(t) + \mathbf{v}(t)\Delta t$$

- This algorithm for "integrating" the equations of motion forward in time in known as Euler's method

Example: A body orbiting the sun

- We consider the Sun's location to be at the origin and the plane of the orbit to be the x-y plane

- In this case we have: $$\mathbf{a}(\mathbf{x}) = \frac{-GM_{\text{sun}}}{r^2}\hat{\mathbf{x}}$$

- Where: $$\hat{\mathbf{x}} = \frac{\mathbf{x}}{r} = \frac{\mathbf{x}}{x^2 + y^2}$$

- The components of the acceleration are then given by:

$$a_x(x, y) = \frac{-GM_{\text{sun}}x}{r^3}, \quad a_y(x, y) = \frac{-GM_{\text{sun}}y}{r^3}$$

Euler's method for body orbiting the sun

- Now we discretize in time and apply Euler's method:

$$v_x(t+\Delta t) = v_x(t) - \frac{GM_{\text{sun}} x(t)\Delta t}{\left(x(t)^2 + y(t)^2\right)^{3/2}}$$

$$v_y(t+\Delta t) = v_y(t) - \frac{GM_{\text{sun}} y(t)\Delta t}{\left(x(t)^2 + y(t)^2\right)^{3/2}}$$

$$x(t+\Delta t) = x(t) + v_x(t)\Delta t$$

$$y(t+\Delta t) = y(t) + v_y(t)\Delta t$$

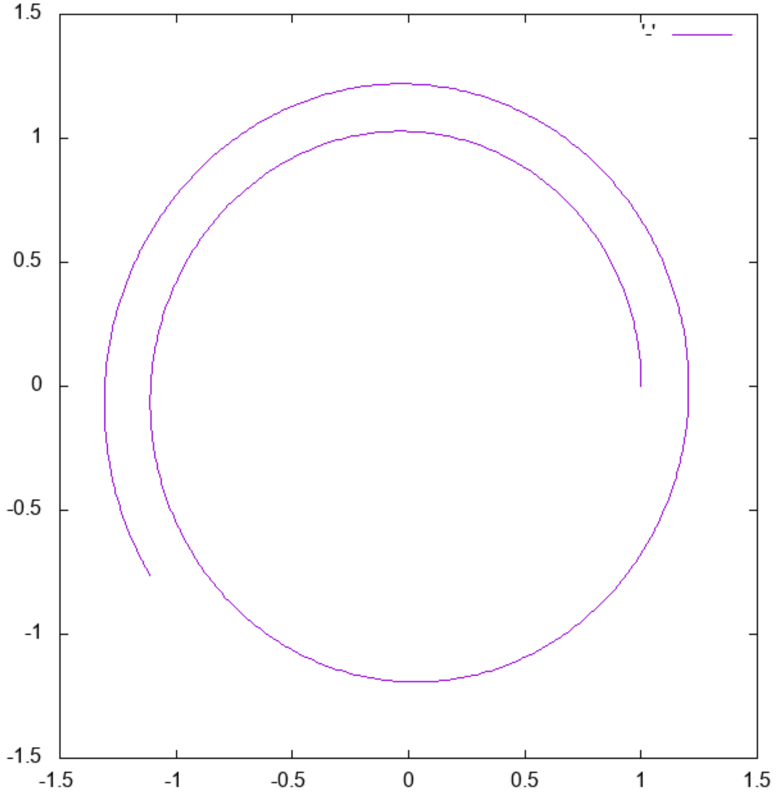
Parameters for orbit problem

- We'll use units of solar masses, and Astronomical Units (AU) for distance
  - In these units, M_{sun}=1 and G = 39.47 $AU^3 M_{sun}^{-1} yr^{-2}$

- Initial conditions:
  - At t = 0 we'll place the body along the x-axis at a distance of 1 AU from the sun and give it the Earth's velocity in the y-direction:
    - $x(0)$ = 1, $y(0)$ =0
    - v_y (0) = 6.283185 AU/yr
  - We will try a time step of 1 day: Δt = 1/365 yr

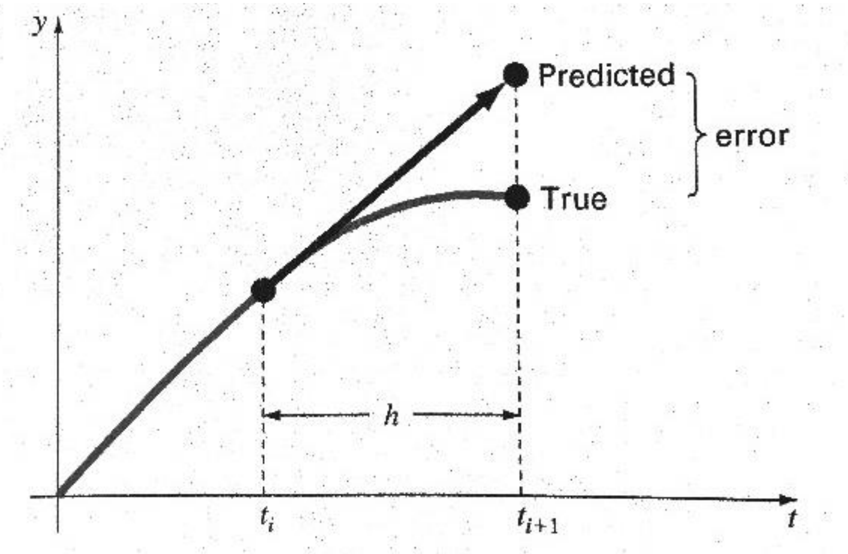
Example program for Euler orbit problem

- See **`orbit_examples.ipynb`**



More accurate ODE numerical methods

- The problem with Euler's method is that the right-hand-side of the equations is evaluated at the beginning of the timestep
- The right-hand-side usually changes over the course of each timestep and we may be getting an inaccurate answer as a result
  - It would be better if we could evaluate the right-hand-side in the middle of the timestep.
  - However, we can't do that unless we know the solution in advance
- We could use higher-order finite differences, however this is not a common approach
- **Strategy:** Use Euler's method to estimate the solution at the midpoint of the timestep. And then use this estimate to evaluate the right-hand-side
- This is called a second order Runge-Kutta method



Aside: Notation for coupled systems of ordinary differential equations

- The equations we were solving with Euler's method were of the form:

$$\frac{dy_1}{dt} = f_1(y_1, y_2, \ldots, y_N, t)$$
$$\frac{dy_2}{dt} = f_2(y_1, y_2, \ldots, y_N, t)$$
$$\vdots$$
$$\frac{dy_N}{dt} = f_N(y_1, y_2, \ldots, y_N, t)$$

- This is a set of coupled first-order ordinary differential equations (ODEs)

Aside: Euler's Method for Coupled Systems of ODEs

- Use shorthand notation for the time at the *n*th step: t^n, and denote $y_i(t^n)$ as y_i^n
- Then approximate the derivatives are written:

$$\frac{dy_i}{dt} \simeq \frac{y_i^{n+1} - y_i^n}{\Delta t}$$

- And Euler's method for a set of coupled ODEs is:

$$\begin{aligned}
y_1^{n+1} &= y_1^n + \Delta t f_1(y_1, y_2, \ldots, y_N, t) \\
y_2^{n+1} &= y_2^n + \Delta t f_2(y_1, y_2, \ldots, y_N, t) \\
&\vdots \\
y_N^{n+1} &= y_N^n + \Delta t f_N(y_1, y_2, \ldots, y_N, t)
\end{aligned}$$

Aside: Coupled systems of ODEs in vector notation

- In order to simplify the description of the second order Runge-Kutta algorithm we use the following vector notation to simplify the equations:

$$\mathbf{y} \equiv (y_1, y_2, y_3, \ldots, y_N)$$

$$\mathbf{f} \equiv (f_1, f_2, f_3, \ldots, f_N)$$

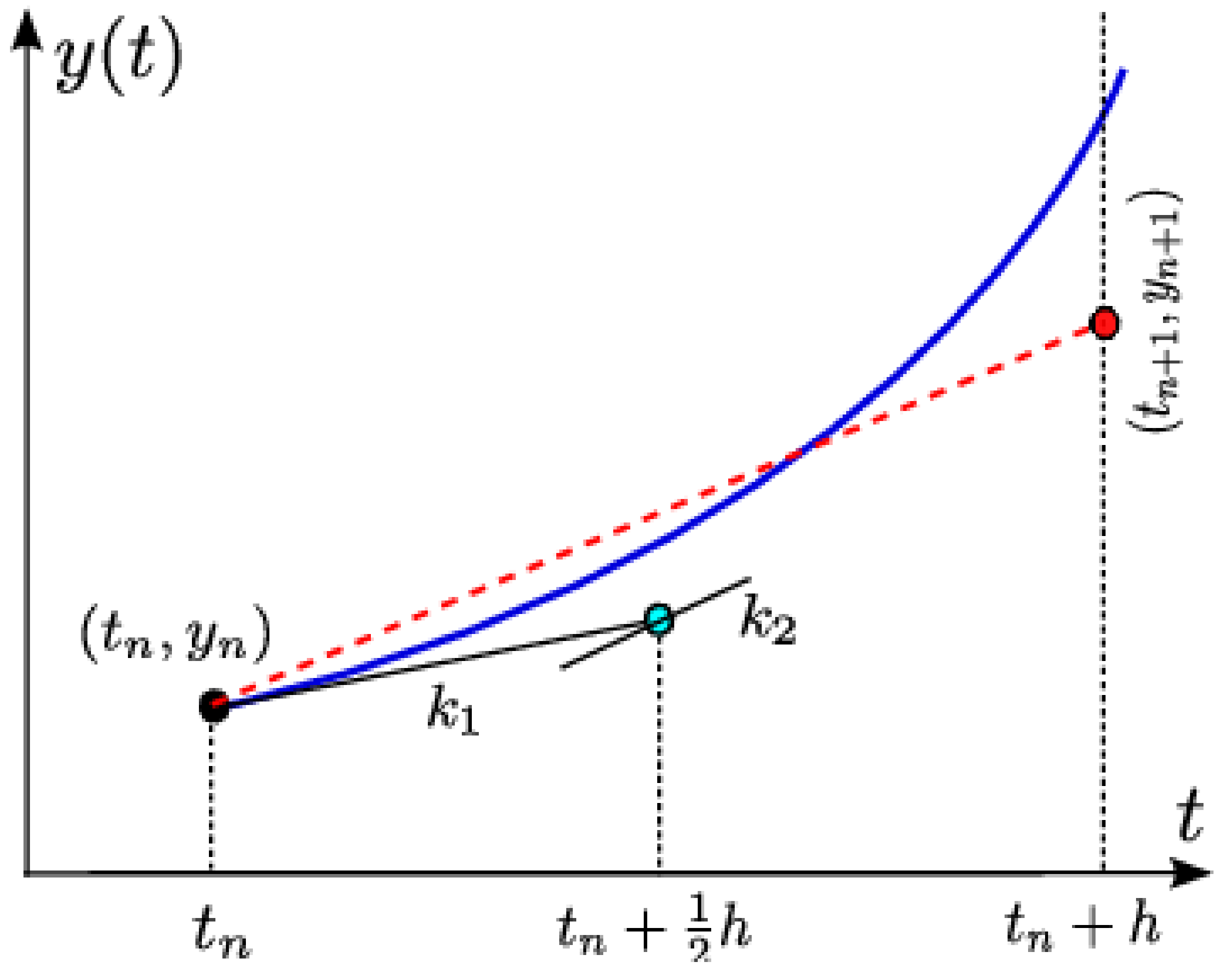
- Using this notation, the original set of ODEs is:

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}, t)$$

- In this notation Euler's method is:

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \Delta t \mathbf{f}(\mathbf{y}^n, t^n)$$

Second-order Runge-Kutta method



Fadlisyah, Muhammad thesis (2014)

Second-order Runge-Kutta method

- Taylor expand around t + 1/2 Δt :

$$y(t+\Delta t) = y(t+\frac{1}{2}\Delta t) + \frac{1}{2}\Delta t \frac{dy}{dt}\bigg|_{t+\frac{1}{2}\Delta t} + \frac{1}{8}\Delta t^2 \frac{d^2y}{dt^2}\bigg|_{t+\frac{1}{2}\Delta t} + \mathcal{O}(\Delta t^3)$$

$$y(t) = y(t+\frac{1}{2}\Delta t) - \frac{1}{2}\Delta t \frac{dy}{dt}\bigg|_{t+\frac{1}{2}\Delta t} + \frac{1}{8}\Delta t^2 \frac{d^2y}{dt^2}\bigg|_{t+\frac{1}{2}\Delta t} - \mathcal{O}(\Delta t^3)$$

- Subtract the two expressions

$$y(t+\Delta t) = y(t) + \Delta t \frac{dy}{dt}\bigg|_{t+\frac{1}{2}\Delta t} + \mathcal{O}(\Delta t^3)$$

Need f evaluated at midpoint

$$= y(t) + \Delta t f(y(t+\frac{1}{2}\Delta t), t+\frac{1}{2}\Delta t) + \mathcal{O}(\Delta t^3)$$

Second-order Runge-Kutta method

- **Step 1:** Estimate change due of the right-hand side using Euler's method:

$$\mathbf{k}_1 = \Delta t \mathbf{f}(\mathbf{y}^n, t^n)$$

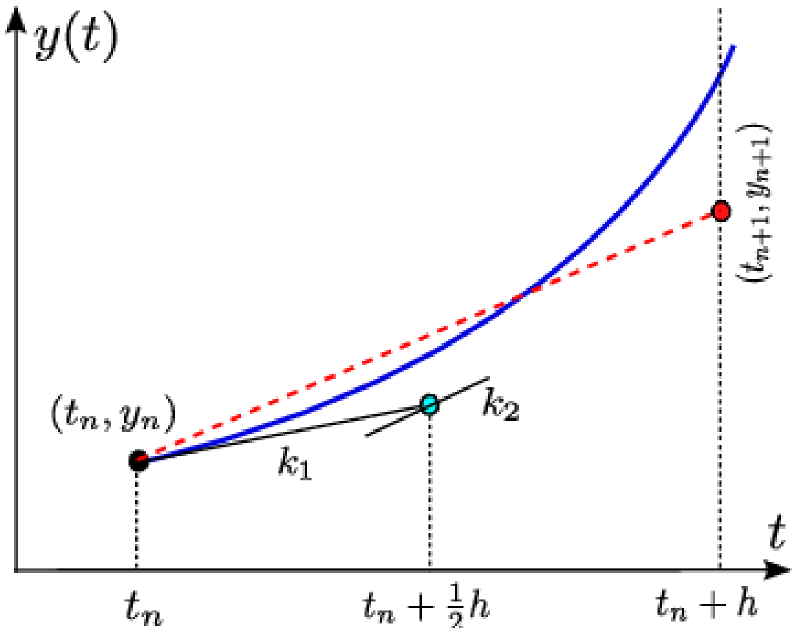
- **Step 2:** Use estimate to predict value of solution at midpoint of the timestep. Evaluate right hand side at midpoint:

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \Delta t \mathbf{f}(\mathbf{y}^n + \frac{1}{2}\mathbf{k}_1, t^n + \frac{1}{2}\Delta t)$$

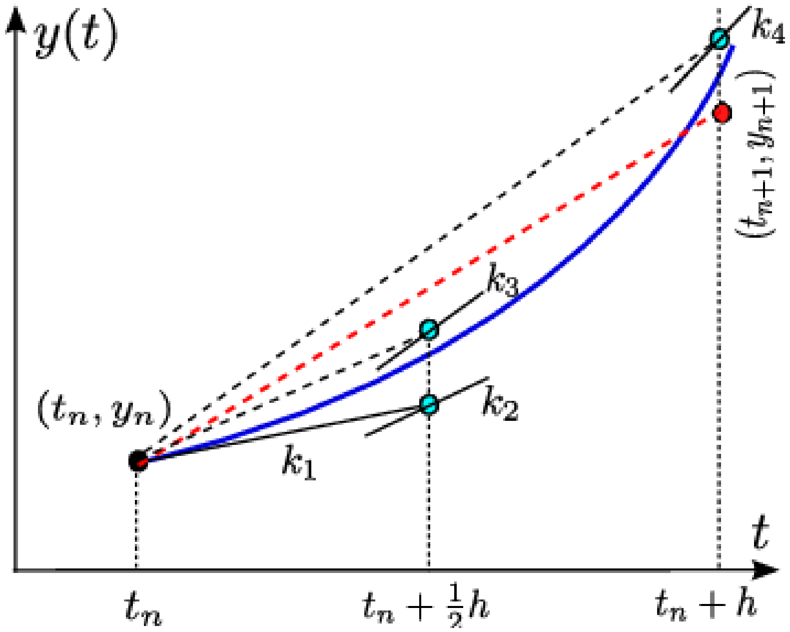
- See **`orbit_examples.ipynb`**

Second and fourth-order Runge-Kutta methods

2nd order



4th order



Fadlisyah, Muhammad thesis (2014)

The fourth-order Runge-Kutta method

- In practice, the workhorse algorithm for first-order sets of ODEs is the fourth-order Runge-Kutta algorithm which (we state here without derivation)

- Step 1: $$\mathbf{k}_1 = \Delta t\ \mathbf{f}(\mathbf{y}^n, t^n)$$

- Step 2: $$\mathbf{k}_2 = \Delta t\ \mathbf{f}(\mathbf{y}^n + \frac{1}{2}\mathbf{k}_1, t^n + \frac{1}{2}\Delta t)$$

- Step 3: $$\mathbf{k}_3 = \Delta t\ \mathbf{f}(\mathbf{y}^n + \frac{1}{2}\mathbf{k}_2, t^n + \frac{1}{2}\Delta t)$$

- Step 4: $$\mathbf{k}_4 = \Delta t\ \mathbf{f}(\mathbf{y}^n + \mathbf{k}_3, t^n + \Delta t)$$

- Step 5: $$\mathbf{y}^{n+1} = \mathbf{y}^n + \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$$

Runge-Kutta methods

- Euler method can be thought of as the first-order RK method
  - Accurate to first order in Δt, i.e., error is order Δt^2

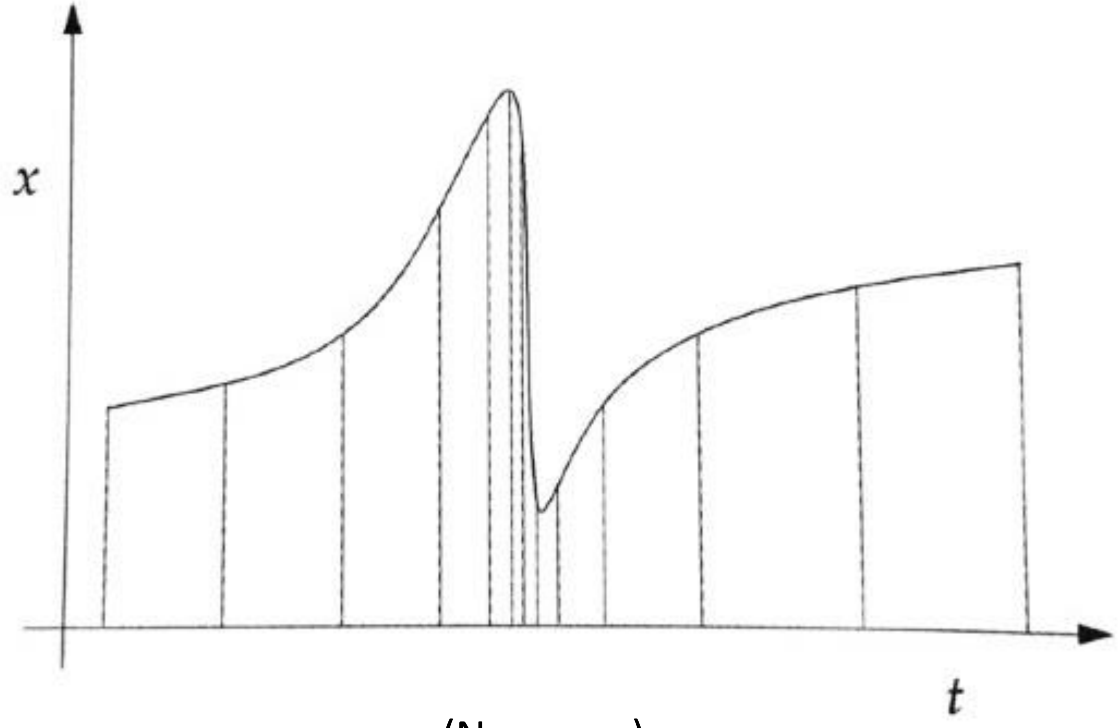
- Second-order RK method accurate to Δt^2, so error Δt^3

- Fourth-order RK method accurate to Δt^4, so error Δt^5
  - By far the most common method for the numerical solution of ODEs
  - Balances accuracy and complexity

- Quoted accuracies are for one step, errors accumulate over the number of steps needed in the calculation, usually loose an order of accuracy (see Newman)

Adaptive step size

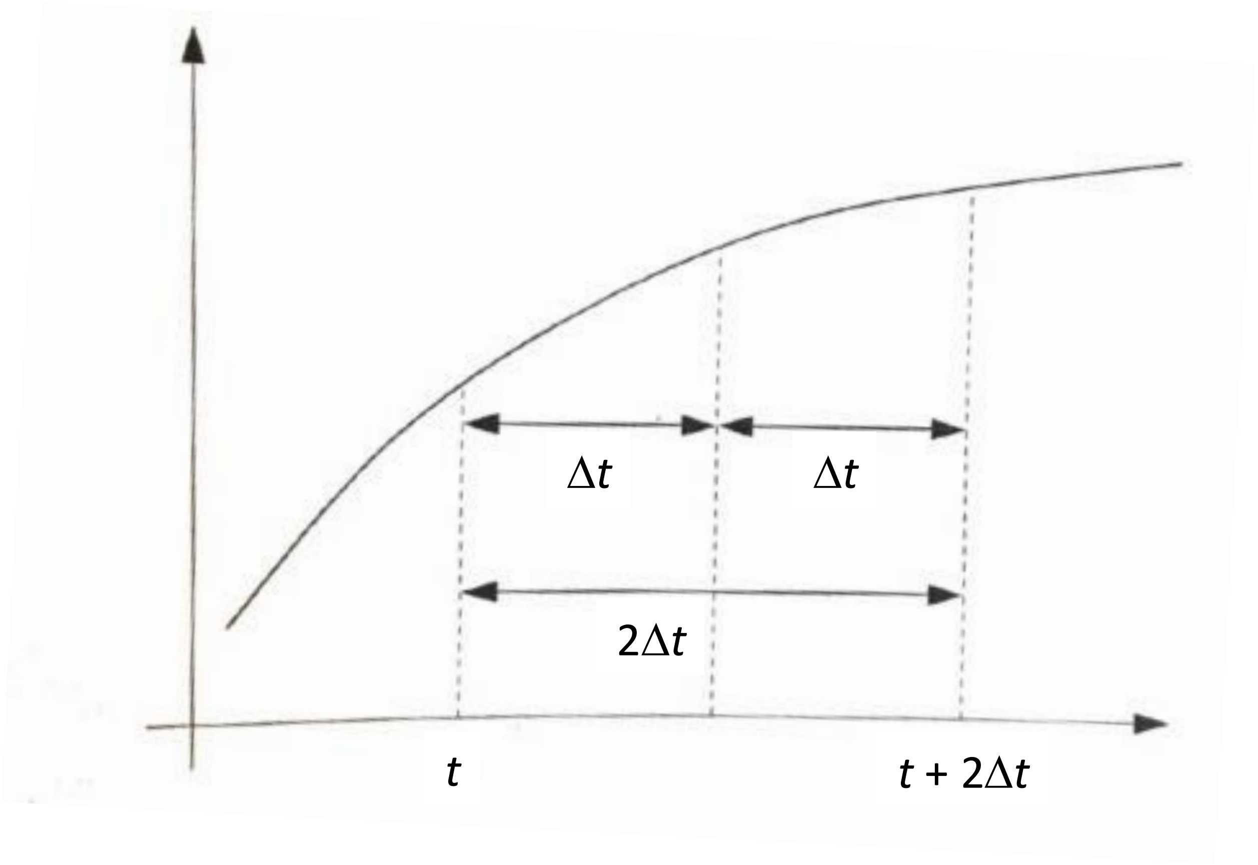
- So far, we have set by hand a constant step size Δt
- Often, we can get better results by varying the step size
  - Increase in regions where function varies rapidly, decrease where it varies slowly
- Approach: vary Δt so the error introduced per unit interval is roughly constant
  - First we need to estimate the error in the steps



(Newman)

Adaptive step size: Estimating the error

- 1. Choose initial (small) Δt
- 2. Use RK method to do two Δt steps of the solution
- 3. Go back to initial t and do an RK step with $2\Delta t$
- 4. Compare the results to estimate the error



Adaptive step size: Estimating the error

- True value of function related to estimate $y_{\Delta t}$:

$$y(t + 2\Delta t) = y_{\Delta t} + 2c\Delta t^5$$

- For doubled step size $y_{2\Delta t}$:

$$y(t + 2\Delta t) = y_{2\Delta t} + 32c\Delta t^5$$

- So per step error is:

$$\epsilon = c\Delta t^5 = \frac{1}{30}(y_{\Delta t} - y_{2\Delta t})$$

- Take δ to be the target accuracy per step. Then the step size necessary to get that accuracy is:

$$\Delta t' = \Delta t \sqrt[5]{\frac{30\delta}{|y_{\Delta t} - y_{2\Delta t}|}}$$

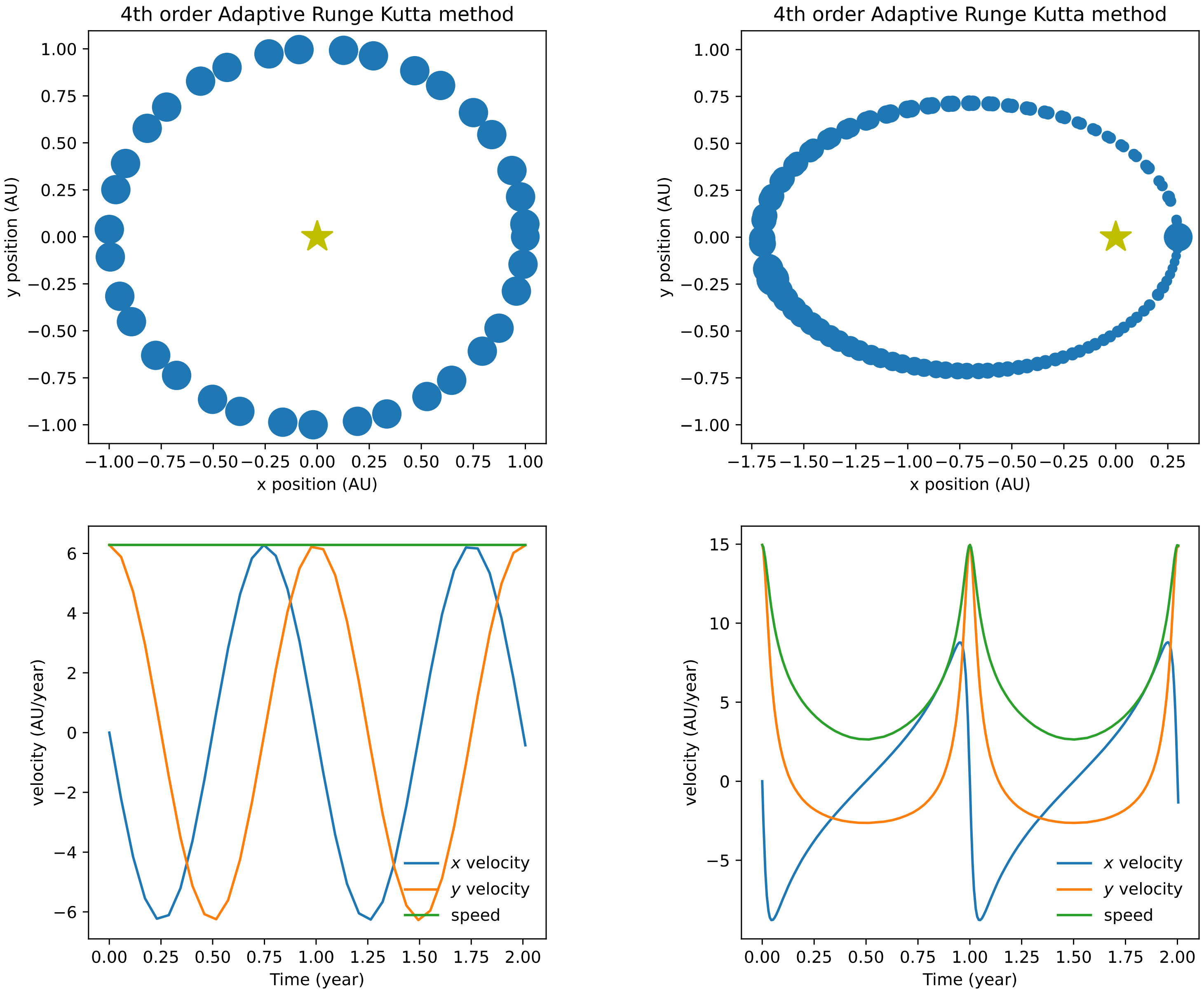
Adaptive step size: Complete approach

- 1. Choose initial Δt
- 2. Use RK method to do two Δt steps of the solution
- 3. Go back to initial t and do an RK step with $2\Delta t$
- 4. Compare the results to estimate the error
- 5. Calculate ideal step size $\Delta t'$
  - If $\varepsilon > \delta$, then redo the calculation with $\Delta t'$
  - If $\varepsilon < \delta$, take the results obtained using Δt and move on to time $t + \Delta t$. In the next iteration use $\Delta t'$ as the timestep

- Requires at least 3 RK steps for every two actually used, but usually results in an overall speedup for a given accuracy
- Usually limit how much $\Delta t'$ can differ from Δt (e.g., by less than a factor of two) in case the denominator happens to diverge

Example: Elliptical orbit with adaptive 4th-order RK

<u>Circular</u>:
x_0= 1 AU
v_{y0} =6.283185 AU/year



<u>Elliptical</u>:
x_0= 0.3 AU
v_{y0} =14.955378 AU/year

Improving the results with local extrapolation

- We can use our knowledge of the error to improve our estimate for $y(t+\Delta t)$ recall that:

$$y(t + 2\Delta t) = y_{\Delta t} + 2c\Delta t^5$$

- And:

$$\epsilon = c\Delta t^5 = \frac{1}{30}(y_{\Delta t} - y_{2\Delta t})$$

- So:

$$y(t + 2\Delta t) = y_{\Delta t} + \frac{1}{15}(y_{\Delta t} - y_{2\Delta t}) + \mathcal{O}(\Delta t^6)$$

- No estimate of the error but presumably better than previous 4th order result

Today's lecture

- Ordinary differential equations:
  - Euler method
  - Runge-Kutta methods and adaptive RK
  - Beyond Runge-Kutta
    - Leapfrog/Verlet/modified midpoint
    - Bulirsch-Stoer Method

Leapfrog method

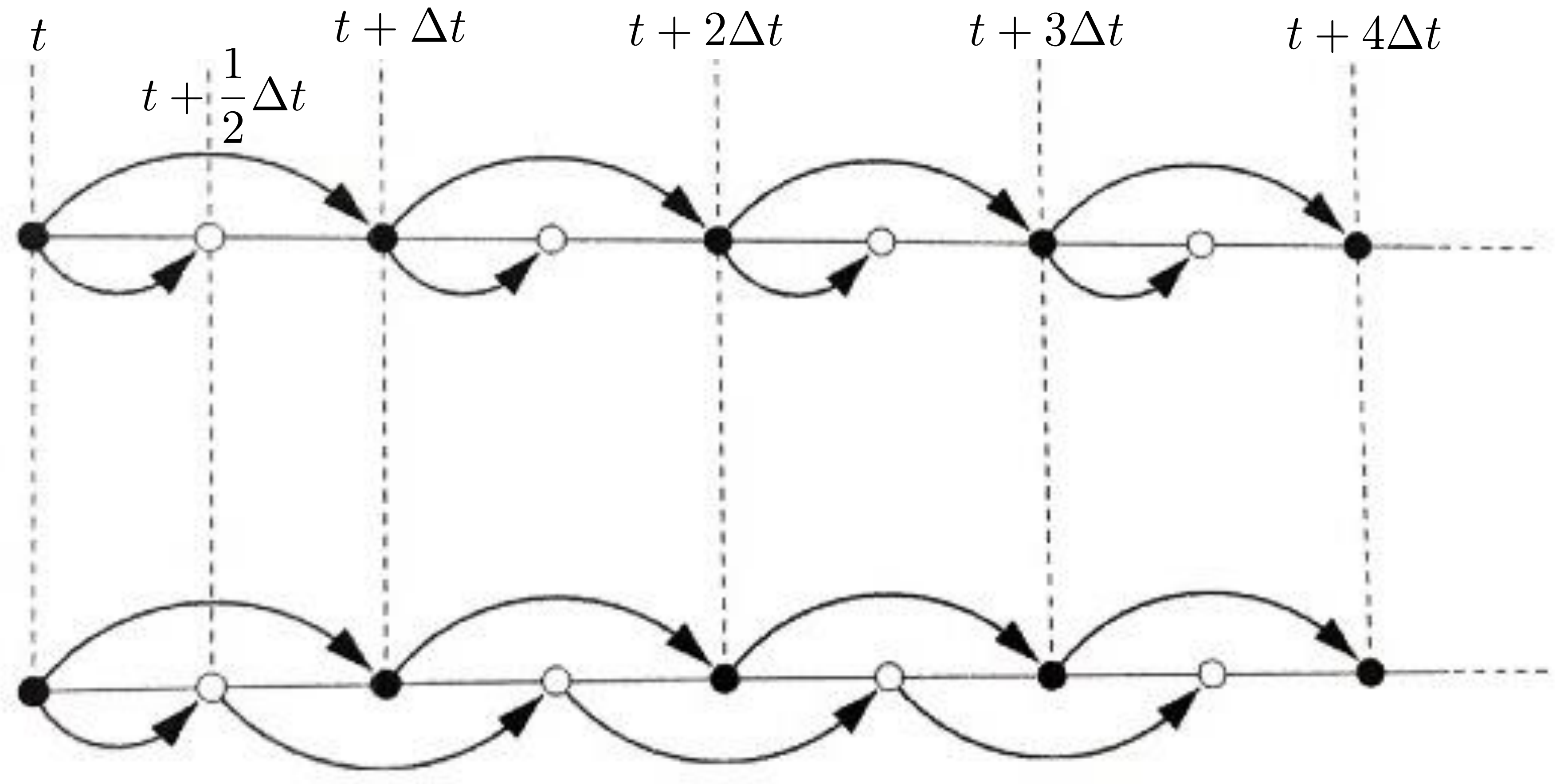
- Recall the second-order RK method:
  - Using the Euler method applied to t to estimate the value of a variable at the midpoint of the interval $t + 1/2\Delta t$

$$y(t + \frac{1}{2}\Delta t) = y(t) + \frac{1}{2}\Delta t f(y, t)$$

$$y(t + \Delta t) = y(t) + \Delta t f\left[y(t + \frac{1}{2}\Delta t), t + \frac{1}{2}\Delta t\right]$$

- Leapfrog method uses a similar approach, except calculates the next midpoint by using the Euler method evaluated at the previous midpoint

Leapfrog method versus 2^{nd} order RK



2^{nd} order RK:

Leapfrog:

(Newman)

Leapfrog method

- Starts out the same as RK:

$$y(t + \frac{1}{2}\Delta t) = y(t) + \frac{1}{2}\Delta t f(y,t)$$

$$y(t + \Delta t) = y(t) + \Delta t f\left[y(t + \frac{1}{2}\Delta t), t + \frac{1}{2}\Delta t\right]$$

- Then:

$$y(t + \frac{3}{2}\Delta t) = y(t + \frac{1}{2}\Delta t) + \Delta t f\left[y(t + \Delta t), t + \Delta t\right]$$

$$y(t + 2\Delta t) = y(t + \Delta t) + \Delta t f\left[y(t + \frac{3}{2}\Delta t), t + \frac{3}{2}\Delta t\right]$$

Why the leapfrog method?

- Time reversal symmetric
  - Useful for physics problems where energy conservation is important
- Error is even in step size
  - Ideal starting point for Richardson extrapolation for Bulirsch-Stoer

Leapfrog method is "time-reversal symmetric"

- If we use $-\Delta t$ instead of Δt, we should retrace our steps
- To see this, start with the equations we repeatedly apply for the Leapfrog method:

$$y(t+\Delta t) = y(t) + \Delta t f\left[y(t+\frac{1}{2}\Delta t), t+\frac{1}{2}\Delta t\right]$$

$$y(t+\frac{3}{2}\Delta t) = y(t+\frac{1}{2}\Delta t) + \Delta t f\left[y(t+\Delta t), t+\Delta t\right]$$

- Set step size to $-\Delta t$:

$$y(t-\Delta t) = y(t) - \Delta t f\left[y(t-\frac{1}{2}\Delta t), t-\frac{1}{2}\Delta t\right]$$

$$y(t-\frac{3}{2}\Delta t) = y(t-\frac{1}{2}\Delta t) - \Delta t f\left[y(t-\Delta t), t-\Delta t\right]$$

Leapfrog method is "time-reversal symmetric"

- Now make a trivial shift in time: $t \to t + \frac{3}{2}\Delta t$
- To get:

$$y(t + \frac{1}{2}\Delta t) = y(t + \frac{3}{2}\Delta t) - \Delta t f\left[y(t + \Delta t), t + \Delta t\right]$$

$$y(t) = y(t + \Delta t) - \Delta t f\left[y(t + \frac{1}{2}\Delta t), t + \frac{1}{2}\Delta t\right]$$

- Same as the original: (but moving backwards)

$$y(t + \Delta t) = y(t) + \Delta t f\left[y(t + \frac{1}{2}\Delta t), t + \frac{1}{2}\Delta t\right]$$

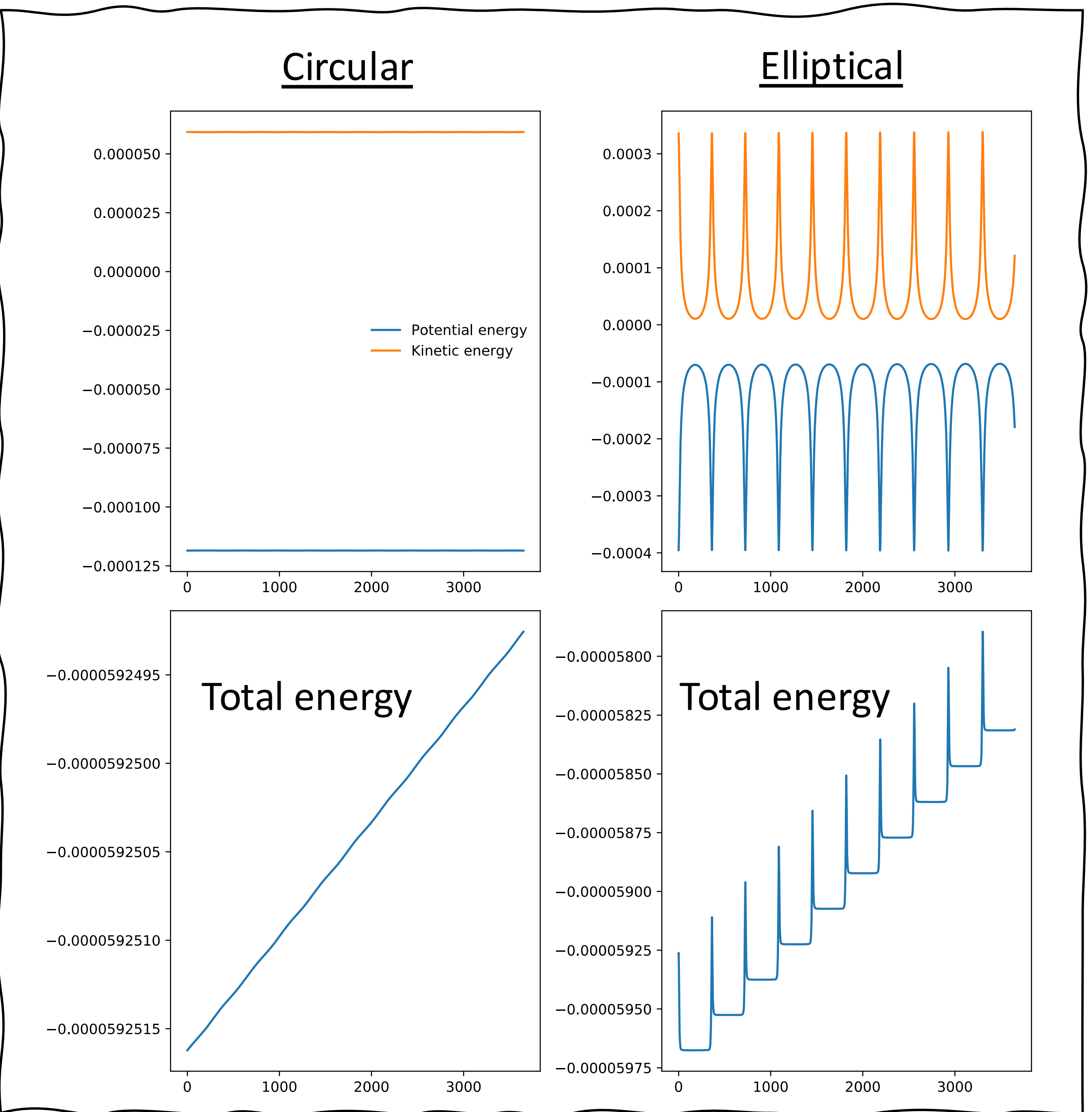
$$y(t + \frac{3}{2}\Delta t) = y(t + \frac{1}{2}\Delta t) + \Delta t f\left[y(t + \Delta t), t + \Delta t\right]$$

What about 2nd order Runge-Kutta?

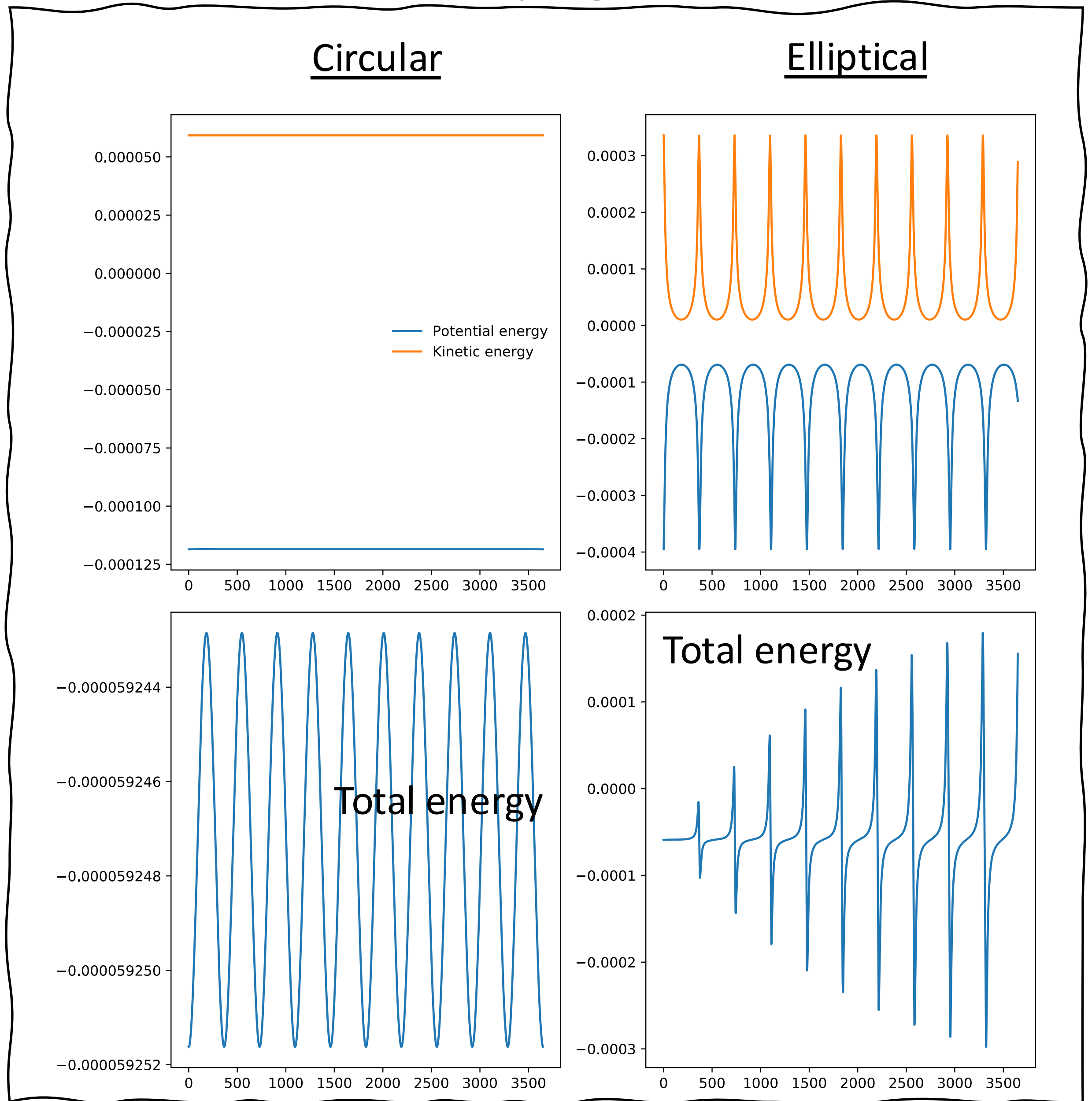
- Original expressions: $$y(t + \frac{1}{2}\Delta t) = y(t) + \frac{1}{2}\Delta t f(y,t)$$
$$y(t + \Delta t) = y(t) + \Delta t f \left[y(t + \frac{1}{2}\Delta t), t + \frac{1}{2}\Delta t \right]$$
- Set step size to $-\Delta t$: $$y(t - \frac{1}{2}\Delta t) = y(t) - \frac{1}{2}\Delta t f(y,t)$$
$$y(t - \Delta t) = y(t) - \Delta t f \left[y(t - \frac{1}{2}\Delta t), t - \frac{1}{2}\Delta t \right]$$
- No way to, e.g., make a shift in *t* to get back to original operations in the opposite direction
  - Errors will result in broken time-reversal symmetry

Why is time-reversal symmetry important? Energy conservation!

2nd order RK



Leapfrog



Verlet method for equations of motion using leapfrog method

- For this method we will limit ourselves to ODEs of the form of equations of motion:

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(t), \qquad \frac{d\mathbf{v}}{dt} = \mathbf{f}(\mathbf{x}, t)$$

- (i.e., where the RHS of the first equation does not depend on **x**)
- In that case, we can do the leapfrog method with two equations

Position only at integer steps

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \mathbf{v}\left(t + \frac{1}{2}\Delta t\right)$$

Velocity only at half-integer steps

$$\mathbf{v}(t + \frac{3}{2}\Delta t) = \mathbf{v}(t + \frac{1}{2}\Delta t) + \Delta t \mathbf{f}\left[\mathbf{x}(t + \Delta t), t + \Delta t\right]$$

What if we want to know, e.g., the total energy at a point?

- Total energy requires knowing **x** and **v** at the same point
- Let's just step the velocity back half a step with Euler's method:

$$\mathbf{v}(t+\frac{1}{2}\Delta t) = \mathbf{v}(t+\Delta t) - \frac{1}{2}\Delta t \mathbf{f}[\mathbf{x}(t+\Delta t), t+\Delta t]$$

- Rearrange to get:

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t+\frac{1}{2}\Delta t) + \frac{1}{2}\Delta t \mathbf{f}[\mathbf{x}(t+\Delta t), t+\Delta t]$$

- Gives velocity at integer points from quantities we have already calculated

Verlet method: Leapfrog in this specific situation of, e.g., EOM:

- First do an initial half step:

$$\mathbf{v}(t+\frac{1}{2}\Delta t)=\mathbf{v}(t)+\frac{1}{2}\Delta t\mathbf{f}\left[\mathbf{x}(t),t\right]$$

- Then repeatedly apply:

$$\mathbf{x}(t+\Delta t)=\mathbf{x}(t)+\Delta t\mathbf{v}\left(t+\frac{1}{2}\Delta t\right)$$

$$\mathbf{k}=\Delta t\mathbf{f}\left[\mathbf{x}(t+\Delta t),t+\Delta t\right]$$

$$\mathbf{v}(t+\Delta t)=\mathbf{v}(t+\frac{1}{2}\Delta t)+\frac{1}{2}\mathbf{k}$$

$$\mathbf{v}(t+\frac{3}{2}\Delta t)=\mathbf{v}(t+\frac{1}{2}\Delta t)+\mathbf{k}$$

After class tasks

- Homework 1 due Sept. 17 (end of the day)
  - Let me know if you have HW questions or questions/issues on github classroom
- Readings:
  - Newman Ch. 8