



PHYS3034 Computational Physics

A/Professor Sveta Postnova

svetlana.postnova@sydney.edu.au

Recap: Week 1

- Errors: *floating point* and *truncation*.
- Simple projectile motion problem.
- Introduced *non-dimensionalisation*, a standard procedure for numerical methods.
- Introduced *Euler's method*, a simple forward difference approximation to the derivative.
 - $\mathbf{r}_{n+1} = \mathbf{r}_n + \tau \mathbf{v}_n$, $\mathbf{v}_{n+1} = \mathbf{v}_n + \tau \mathbf{a}_n$.

Computer Lab (Lab 1)

- Evaluated how local and global errors scale with τ .
- Euler's method worked ok!
- Implemented(!) the *midpoint method*, which provides an exact numerical method for the constant acceleration case.
 - $\mathbf{v}_{n+1} = \mathbf{v}_n + \tau \mathbf{a}_n$, $\mathbf{r}_{n+1} = \mathbf{r}_n + \frac{1}{2} \tau (\mathbf{v}_n + \mathbf{v}_{n+1})$
 - Requires the velocity update to be done before the position update.

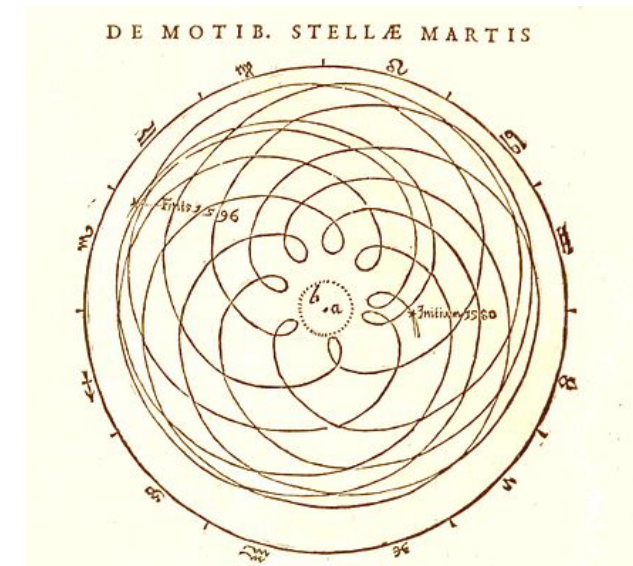
Lecture 2: Verlet integration

- ✓ Euler method works well for simple, non-stiff, slowly varying systems over short times
- ✗ Euler performs poorly for stiff, oscillatory, chaotic, or long-time Hamiltonian systems
Examples: harmonic oscillator, Lorenz system, orbital motion

Today's example: **the Kepler Problem**

- Special case of the two-body problem
- Orbital motion

Check out some orbits around Earth.



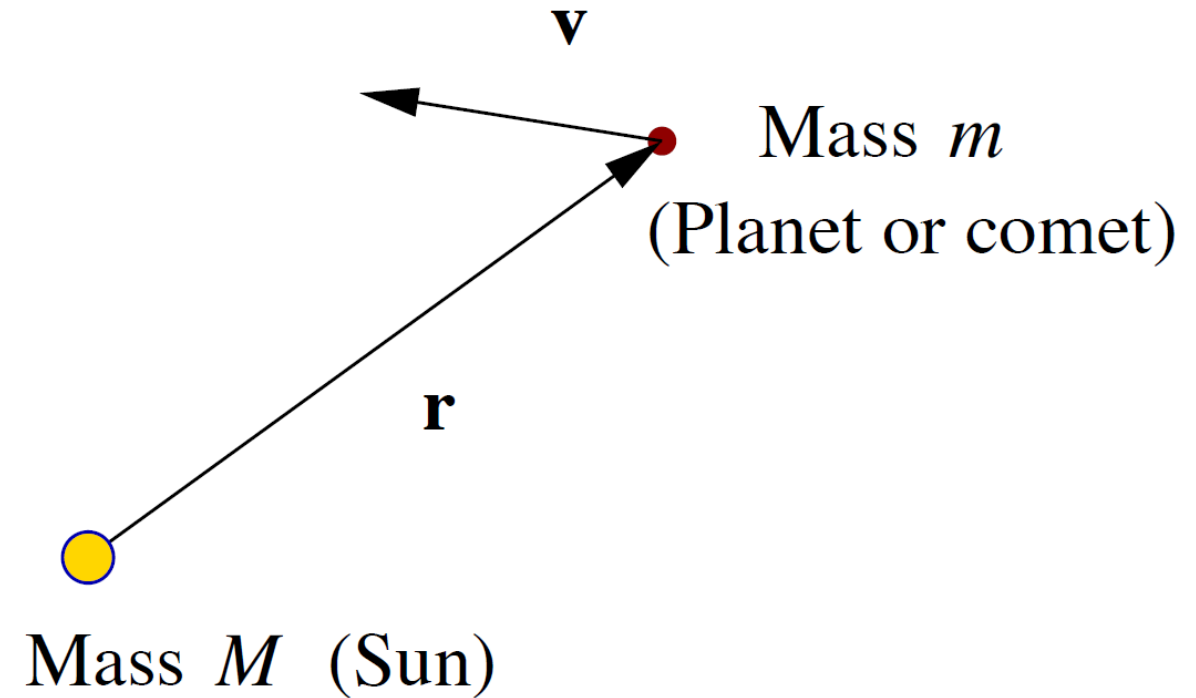
Lecture 2: Outline

- The Kepler problem as a *dynamical system*.
- *Non-dimensionalisation* of the Kepler problem.
- Euler's method (forward-difference) is unsatisfactory.
- *Centred-difference approximation*.
 - Yields a new method: *Verlet integration*.
 - Accurately integrates Keplerian orbits with near-conservation of total energy!
- *Codes*: kepler_verlet.ipynb,
kepler_dynamics.ipynb



The Kepler problem

- Dynamics of some object (e.g., a planet) of mass, m , under the gravitational force of the Sun.
- Assume the Sun (mass $M \gg m$) is fixed at the origin.
- Paths followed are called *orbits*.
- Stable orbits trace out ellipses with the Sun at a focus.



Preparing to solve the Kepler Problem numerically

1. Define the Kepler problem as a dynamics problem
2. Identify testing scenario(s)
3. Non-dimensionalise the system for easier numerical handling

After that, we can solve it with whichever method we choose

1. The Kepler problem as a dynamics problem

Recall: Dynamics problems

$$\frac{d\mathbf{r}}{dt} = \mathbf{v}, \quad \frac{d\mathbf{v}}{dt} = \mathbf{a}(\mathbf{r}, \mathbf{v}, t).$$

- Describe the motion of particle(s).
- Numerical methods (e.g., Euler) allow us to step forward in time (time step of size τ) and piece together an approximate trajectory:
 - $(\mathbf{r}_1, \mathbf{v}_1) \xrightarrow{\tau} (\mathbf{r}_2, \mathbf{v}_2) \xrightarrow{\tau} \dots$
 - Local truncation error in each step.

Gravitational acceleration and the Kepler problem

A particle of mass m moves under Newtonian gravity: $\mathbf{F} = -\frac{GMm}{r^2}\hat{\mathbf{r}}$

$$G \approx 6.67 \times 10^{-11} \text{ kg}^{-1}\text{m}^3\text{s}^{-2}.$$

The equation of motion is then: $m \frac{d^2\mathbf{r}}{dt^2} = -\frac{GMm}{r^2}\hat{\mathbf{r}} \rightarrow \mathbf{a}(\mathbf{r}) = -\frac{GM}{r^2}\hat{\mathbf{r}} = -\frac{GM}{|\mathbf{r}|^3}\mathbf{r}$

The acceleration depends only on position \mathbf{r} , and points towards the origin (Newton, 1687)

So the *Keplerian equations of motion*:

$$\boxed{\frac{d\mathbf{r}}{dt} = \mathbf{v}, \quad \frac{d\mathbf{v}}{dt} = -\frac{GM}{r^2}\hat{\mathbf{r}}}$$

- Equivalent second-order ODE: $\frac{d^2\mathbf{r}}{dt^2} = -\frac{GM}{r^3}\mathbf{r}$

2. The Kepler problem: test case

Solving the Kepler problem

$$\frac{d\mathbf{r}}{dt} = \mathbf{v}, \quad \frac{d\mathbf{v}}{dt} = -\frac{GM}{r^2} \hat{\mathbf{r}}$$

Our goals:

1. *Solve these ODEs* to numerically compute planetary orbits.
 - Given some initial conditions $\mathbf{r}(0) = \mathbf{r}_1$, and $\mathbf{v}(0) = \mathbf{v}_1$
2. *Assess the accuracy* of numerical solutions against known quantities:
 - Test the *conservation* of physical quantities that we know should be conserved.
 - Angular momentum: $\mathbf{L} = \mathbf{r} \times m\mathbf{v}$
 - Total energy: $E = \frac{1}{2}mv^2 - \frac{GMm}{r}$
 - [**Exercise**: Prove $\frac{dL}{dt} = 0$ and $\frac{dE}{dt} = 0$].
 - Compare the simulated trajectory against an *analytic solution*.


Special case for testing: the Circular Motion Solution

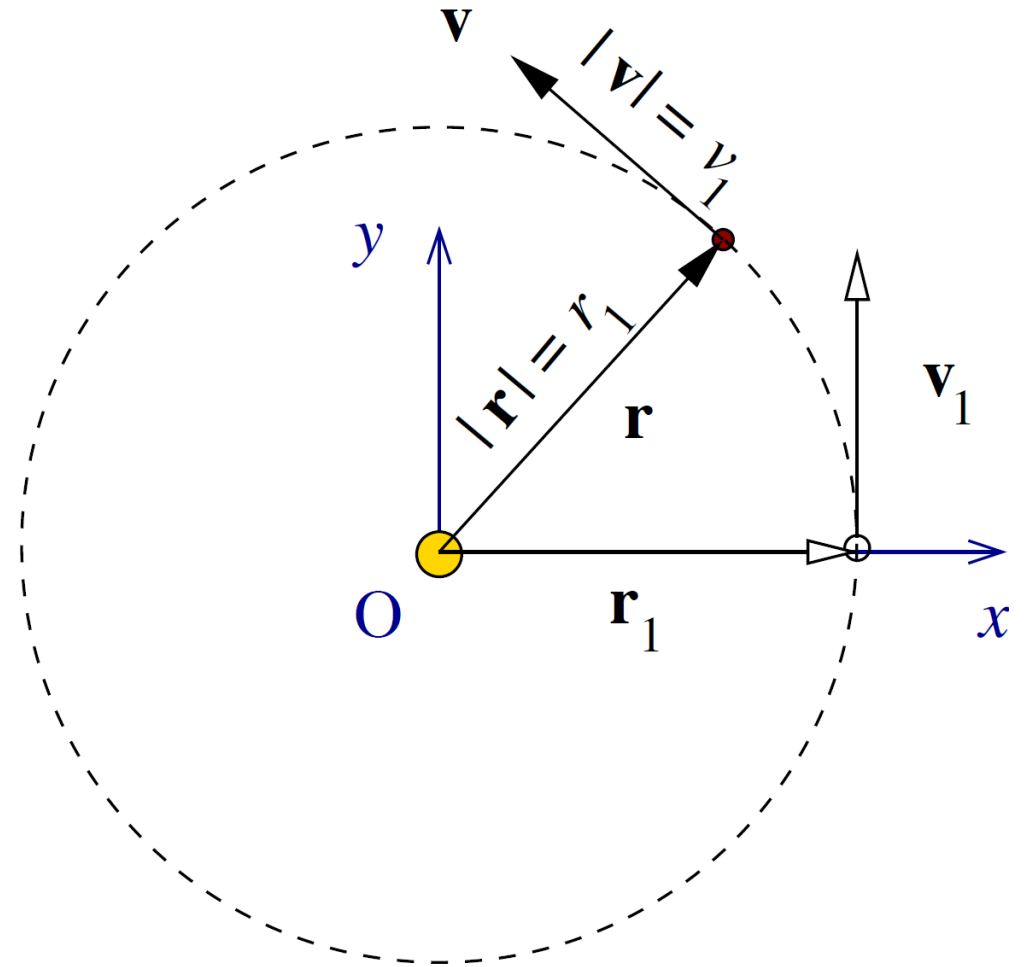
- We can parametrize the *circular motion solution* as $\mathbf{r}_c(t) = r_1 [\cos(\omega t), \sin(\omega t)]$.
 - Can solve (e.g., using 2nd-order ODE) for the angular frequency, $\omega = \omega_c = \frac{(GM)^{1/2}}{r_1^{3/2}}$.
 - And so the period, $T_c = \frac{2\pi}{\omega} = \frac{2\pi r_1^{3/2}}{(GM)^{1/2}}$.
- Circular motion is a *good test case* to evaluate numerical methods
 - Because it's easy to see when we wobble off course 🙈
- We should obtain circular motion from $\mathbf{r}(0) = (r_1, 0)$ and $\mathbf{v}(0) = (0, v_1)$, with $v_1 = \omega r_1 = \sqrt{\frac{GM}{r_1}}$.
 - **Exercise:** Show that these choices reproduce the requirement $\mathbf{r}_c(t) = r_1 [\cos(\omega t), \sin(\omega t)]$.

Circular Motion

- We start at $\mathbf{r}(0) = (r_1, 0)$ and $\mathbf{v}(0) = (0, v_1)$.


- $v_1 = \omega r_1 = \sqrt{\frac{GM}{r_1}}$.

- This is our circular motion test solution .



3. Non-dimensionalisation of the Kepler problem

Non-dimensionalisation

- We *rescale variables* to form new dimensionless versions by dividing by dimensional constants $\bar{\mathbf{r}} = \frac{\mathbf{r}}{L_s}$, $\bar{t} = \frac{t}{t_s}$, $\bar{\mathbf{v}} = \frac{\mathbf{v}}{L_s/t_s}$.
- Rewriting the second-order equation of motion: $\frac{d^2 \mathbf{r}}{dt^2} = -\frac{GM}{r^3} \mathbf{r} \Rightarrow \frac{d^2 \bar{\mathbf{r}}}{d\bar{t}^2} = -\frac{GM t_s^2}{L_s^3} \frac{\bar{\mathbf{r}}}{\bar{r}^3}$
 - ? What's a convenient choice for t_s ?
 - 👁 $t_s = \sqrt{\frac{L_s^3}{GM}}$, which gives $\frac{d^2 \bar{\mathbf{r}}}{d\bar{t}^2} = -\frac{\hat{\mathbf{r}}}{\bar{r}^2} = -\frac{\bar{\mathbf{r}}}{\bar{r}^3}$
 - So we get $\frac{d\bar{\mathbf{r}}}{d\bar{t}} = \bar{\mathbf{v}}, \quad \frac{d\bar{\mathbf{v}}}{d\bar{t}} = -\frac{\bar{\mathbf{r}}}{\bar{r}^3}.$
 - Thanks non-dimensionalisation: no more G and M to worry about 

Non-dimensionalisation for total energy (must be conserved)

$$E = \frac{1}{2}mv^2 - \frac{GMm}{r}.$$

- We can also write down a non-dimensional form for the total energy, $\bar{E} = \frac{E}{E_s}$.
- A natural choice: $E_s = \frac{GMm}{L_s}$
- Yields $\bar{E} = \frac{1}{2}\bar{v}^2 - \frac{1}{\bar{r}}$.

Non-dimensional for the circular motion test

- We have $\bar{v}_1 = \bar{r}_1^{-1/2}$, $\bar{\omega}_c = t_s \omega_c = \bar{r}_1^{-3/2}$, and $\bar{T}_c = 2\pi \bar{r}_1^{3/2}$.
- If we set the characteristic length scale for the problem, $L_s = r_1$ (the initial distance of the body from the origin), then...
 - Initial conditions: $\mathbf{r}(0) = (1, 0)$ and $\mathbf{v}(0) = (0, 1)$ yield *circular motion* with $r_c = 1$.
 - Orbits trace out the *unit circle* with period $T_c = 2\pi$.

Numerical solution #1: Euler method

Euler method for the Kepler problem

kepler_dynamics.ipynb

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \tau \mathbf{v}_n ,$$

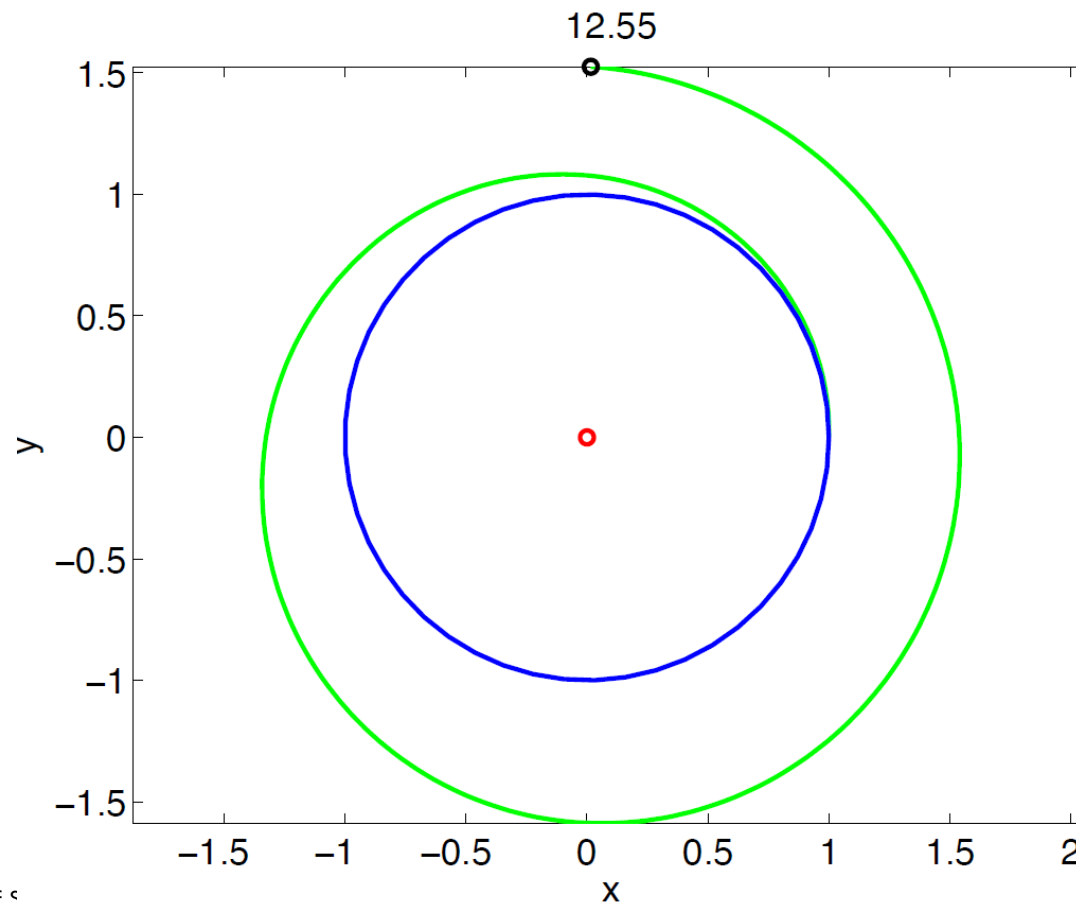
$$\mathbf{v}_{n+1} = \mathbf{v}_n + \tau \mathbf{a}_n .$$

- Time steps forward discretely, as $t_n = (n - 1)\tau$
- The Euler method (forward-difference approximation) worked ok for projectile motion...
 - What about for the Kepler problem?!

- Euler scheme for the Kepler problem is implemented in: kepler_dynamics.ipynb:
 - Integrates for a total time 4π (should be two orbits).
 - Default (non-dimensional) time step is $\tau = 0.05$
 - Code plots the orbit and the analytic solution (unit circle).
 - Calculates non-dimensional $E(t)$ at each time step.
- Let's test it!

Euler on Kepler

- *It doesn't go well at all !*
 - The orbit spirals out and the total energy increases.
 - Smaller time steps, τ , only delay the inevitable.

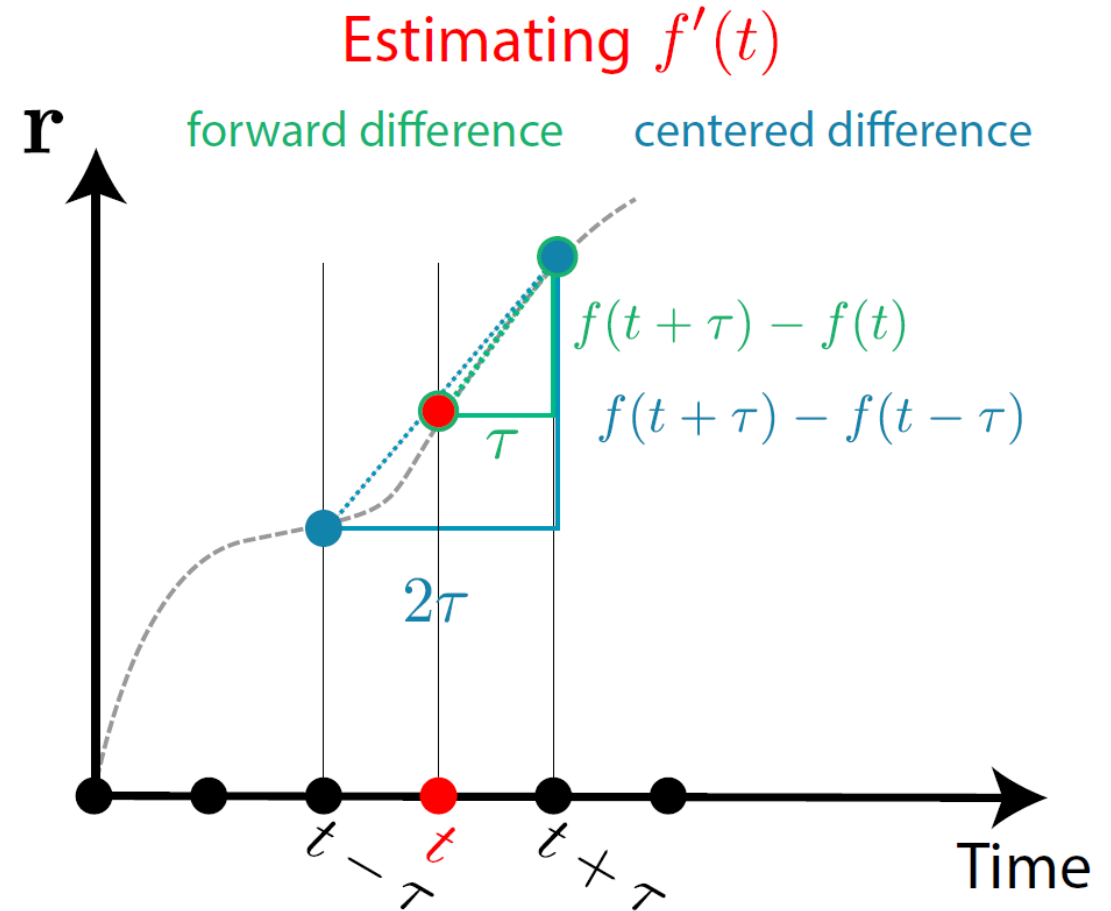


Without testing against known cases, we never know whether we're off track!

A different approach: **Centred difference approximation**

Centred difference approximation

- Euler uses the *forward difference approximation*:
 - $f'(t) = \frac{f(t + \tau) - f(t)}{\tau} + O(\tau).$
- Consider an alternative: the *centered difference approximation*
 - $f'(t) = \frac{f(t + \tau) - f(t - \tau)}{2\tau} + O(\tau^2).$
- **?** Does the higher-order truncation error— $O(\tau^2)$ instead of $O(\tau)$ —mean more or less accurate?



Derivation: the centred difference approximation

- Comes from manipulating the Taylor series:
 - $f(t \pm \tau) = f(t) \pm \tau f'(t) + \frac{1}{2!} \tau^2 f''(t) \pm \frac{1}{3!} \tau^3 f^{(3)}(t) + \dots$
- Even powers of τ cancel when you compute $f(t + \tau) - f(t - \tau)$.
- We get $f'(t) = \frac{f(t + \tau) - f(t - \tau)}{2\tau} - \frac{1}{6} \tau^2 f^{(3)}(t) + \dots$
 - Which we can write as $f'(t) = \frac{f(t + \tau) - f(t - \tau)}{2\tau} + O(\tau^2)$
- Truncation errors $O(\tau^2)$ are pretty good 😊

Centred difference approximation: 2nd derivative

$$f''(t) = \frac{f(t + \tau) - 2f(t) + f(t - \tau)}{\tau^2} + O(\tau^2).$$

- A workhorse in this unit! 🐎🐎🐎
 - (we use this alot)

A sketch on how to get it

- Decompose the second derivative as: $\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial x} \right)$.
- For the first derivative: let $g(x) = \frac{\partial f}{\partial x} \approx \frac{f(x) - f(x - h)}{h}$
- We can do a forward difference approximation to g as: $\frac{\partial g}{\partial x} \approx \frac{g(x + h) - g(x)}{h}$.
- Putting them together:
$$\frac{\partial^2 f}{\partial x^2} \approx \frac{f(x + h) - f(x)}{h^2} - \frac{f(x) - f(x - h)}{h^2} = \frac{f(x + h) - 2f(x) + f(x - h)}{h^2}$$
- **Exercise:** Derive it fully from the Taylor-series expansion
 - By expanding about x at $f(x + h)$ and $f(x - h)$.

Numerical solution #2: Verlet method

based on Centred difference approximation

Centred difference approximation for dynamics

- *Dynamics equations:* $\frac{d^2 \mathbf{r}}{dt^2} = \mathbf{a}$ and $\frac{d\mathbf{r}}{dt} = \mathbf{v}$.
- Apply centered difference approximations to $\left. \frac{d^2 \mathbf{r}}{dt^2} \right|_{t=t_n}$ and $\left. \frac{d\mathbf{r}}{dt} \right|_{t=t_n}$
 - For $t_n = (n-1)\tau$.
- Yields the *Verlet Update Equations:*
 - $\mathbf{r}_{n+1} = 2\mathbf{r}_n - \mathbf{r}_{n-1} + \tau^2 \mathbf{a}_n + O(\tau^4)$
 - $\mathbf{v}_n = \frac{\mathbf{r}_{n+1} - \mathbf{r}_{n-1}}{2\tau} + O(\tau^2).$

Updating using the Verlet method

$$\mathbf{r}_{n+1} = 2\mathbf{r}_n - \mathbf{r}_{n-1} + \tau^2 \mathbf{a}_n, \quad \mathbf{v}_n = \frac{\mathbf{r}_{n+1} - \mathbf{r}_{n-1}}{2\tau}.$$

- *Notice*: You need to update \mathbf{r} before you can update \mathbf{v} .
- *Getting started*: Consider the first step, $n = 1$ for \mathbf{r} . This requires $\mathbf{r}_{n-1} = \mathbf{r}(-\tau)$: a value *before the initial condition*?! 😊
- *Solution*: we can remove \mathbf{r}_{n-1} from the Verlet method equations so that
$$\mathbf{r}_{n+1} = \mathbf{r}_n + \tau \mathbf{v}_n + \frac{1}{2} \tau^2 \mathbf{a}_n.$$
- (see also the *midpoint method* from Lab 1, Q2).
- So we can get started (for $n = 1$) using this midpoint equation, and then apply the Verlet update equations for all remaining time steps, $n \geq 2$. 🏃🏃🏃

We can simulate dynamics without computing \mathbf{v}

$$\mathbf{r}_{n+1} = 2\mathbf{r}_n - \mathbf{r}_{n-1} + \tau^2 \mathbf{a}_n, \quad \mathbf{v}_n = \frac{\mathbf{r}_{n+1} - \mathbf{r}_{n-1}}{2\tau}.$$

- If $\mathbf{a} = \mathbf{a}(\mathbf{r})$, the Verlet update equations allow us to evolve \mathbf{r} *without ever calculating* \mathbf{v} :
 - $\mathbf{r}_1 \xrightarrow{\tau} \mathbf{r}_2 \xrightarrow{\tau} \mathbf{r}_3 \xrightarrow{\tau} \dots$
- Advantageous if we only want to solve for \mathbf{r} 😊

Verlet method: Truncation errors

$$\mathbf{r}_{n+1} = 2\mathbf{r}_n - \mathbf{r}_{n-1} + \tau^2 \mathbf{a}_n + O(\tau^4), \quad \mathbf{v}_n = \frac{\mathbf{r}_{n+1} - \mathbf{r}_{n-1}}{2\tau} + O(\tau^2).$$

- The error terms in the Verlet method equations are deceptive.
- Successive updates are not independent (each update uses the current, \mathbf{r}_n and previous, \mathbf{r}_{n-1} values).
 - 🔥 So you need to write *two successive updates in the form of Taylor expansions* to correctly identify the local truncation errors.
- They are $O(\tau^3)$ for both position and velocity.
 - Global error as $O(\tau^2)$: a second-order method.
- Given its simplicity, this method is quite accurate! 😊

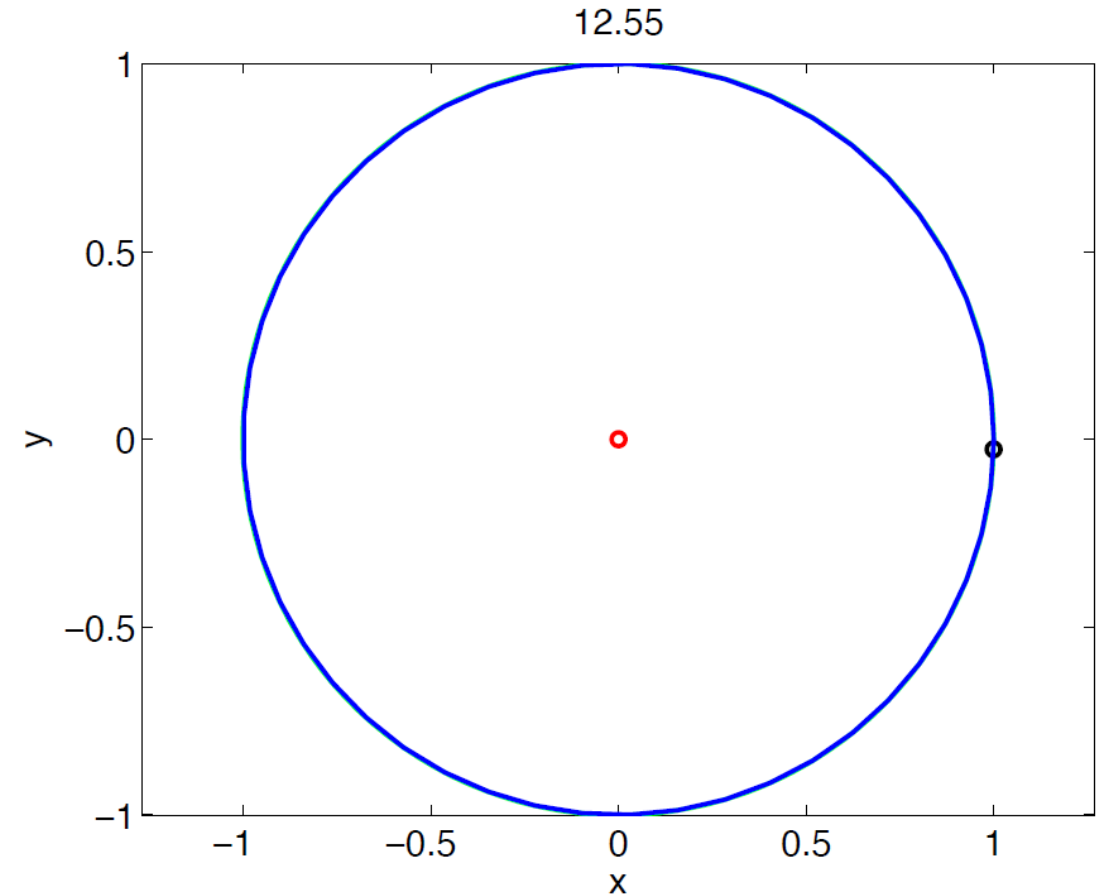
An alternative formulation: Velocity-Verlet

- Another (equivalent) way of writing the Verlet updates is called *Velocity-Verlet*:
 - $\mathbf{r}_{n+1} = \mathbf{r}_n + \tau \mathbf{v}_n + \frac{1}{2} \tau^2 \mathbf{a}_n$,
 $\mathbf{v}_{n+1} = \mathbf{v}_n + \frac{1}{2} \tau (\mathbf{a}_n + \mathbf{a}_{n+1})$.
- The position update is the midpoint method (cf. Computer Lab 1, Q2).
- Requires velocity to be calculated (the $\tau \mathbf{v}_n$ term).
- **Exercise:** Derive this scheme.

Verlet solution to the Kepler problem

kepler_verlet.ipynb

- For $n = 1$ (midpoint method to get things running 🏃), use $\mathbf{r}_{n+1} = \mathbf{r}_n + \tau \mathbf{v}_n + \frac{1}{2} \tau^2 \mathbf{a}_n$.
- Then standard Verlet updates for $n > 1$:
$$\mathbf{r}_{n+1} = 2\mathbf{r}_n - \mathbf{r}_{n-1} + \tau^2 \mathbf{a}_n, \quad \mathbf{v}_n = \frac{\mathbf{r}_{n+1} - \mathbf{r}_{n-1}}{2\tau}.$$
- Accurately integrates the circular test case using $\tau = 0.05$!
- Energy is (nearly) conserved
 - (oscillates $\sim 10^{-7}$)!! COOL COOL COOL



Let's test

Computer Lab 2

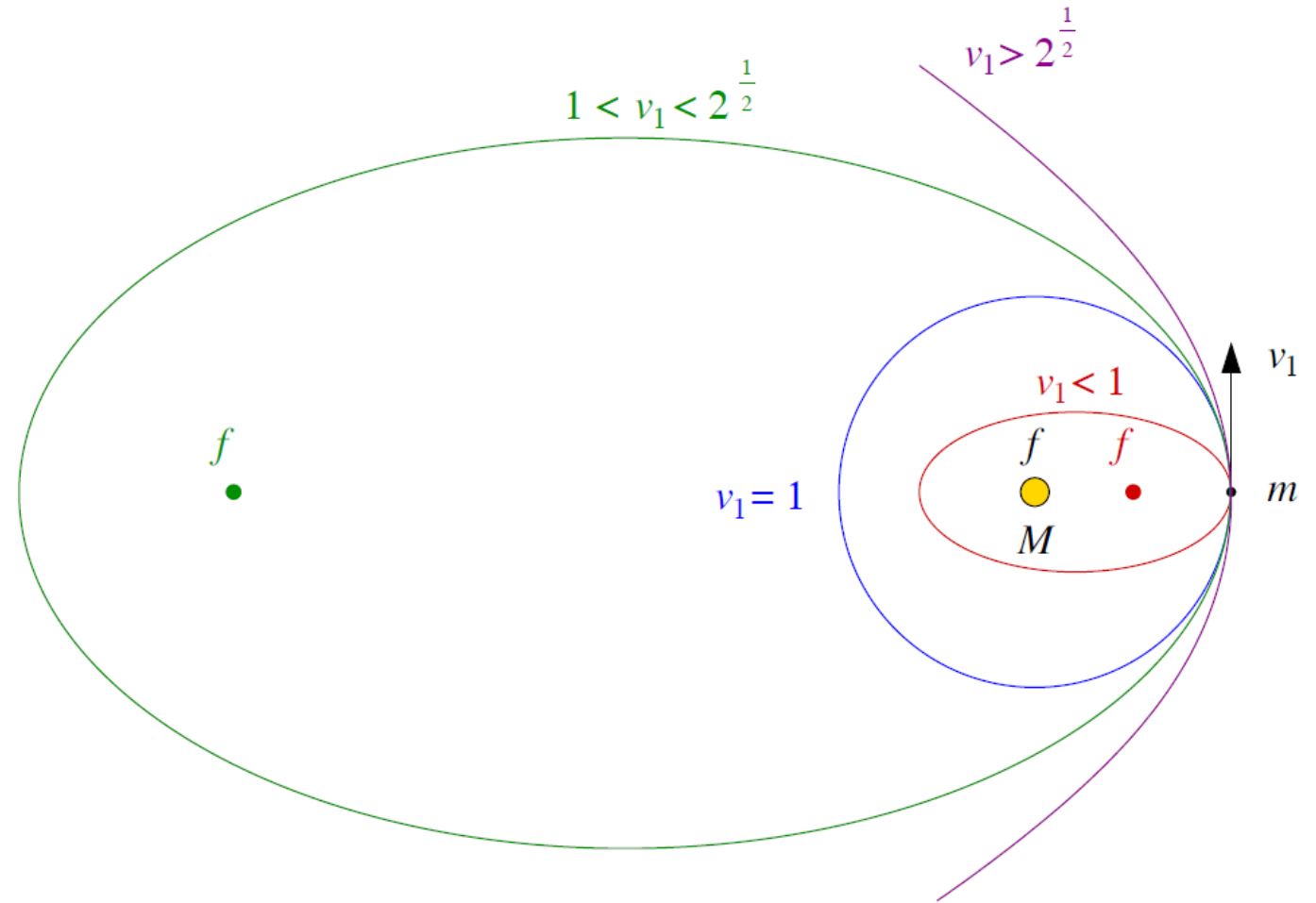
Solutions to the Kepler problem: ellipses and hyperbolae

For initial conditions

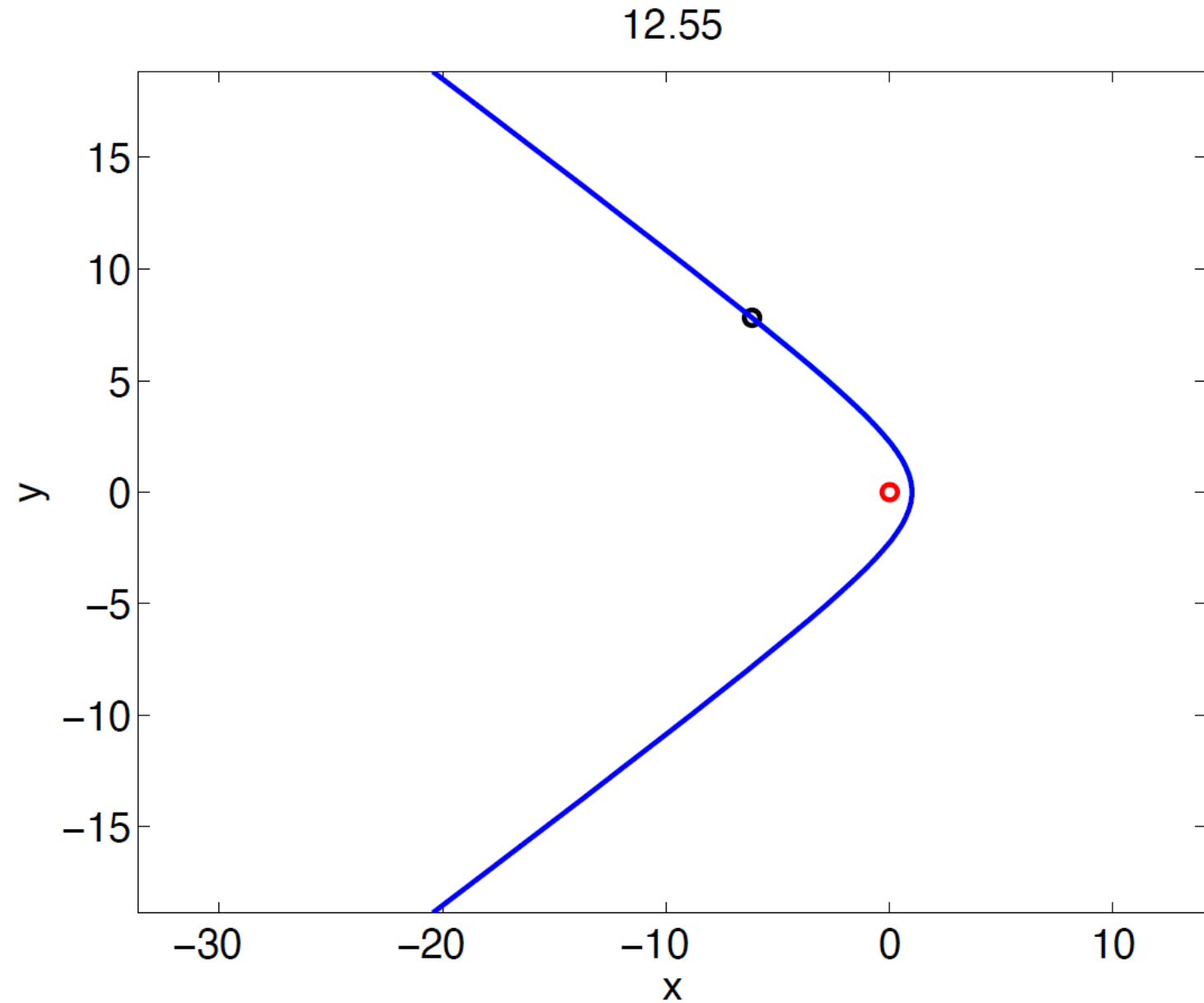
$$\mathbf{r}_1 = (1, 0)$$

$$\mathbf{v}_1 = (0, v_1)$$

we can get different types of solutions depending on v_1 :



Example: numerical solution for $v_1 = 1.5$; a hyperbolic orbit.



Appendix: Analytic solutions to the Kepler problem

- Nice to compare the performance of numerical methods to analytic solution: `kepler_analytic`.
- *Cartesian* form of (elliptic) solution for $\mathbf{r}_1 = (1, 0)$ (closest point of approach) and $\mathbf{v}_1 = (0, v_1)$:
 - $\frac{(x + ae)^2}{a^2} + \frac{y^2}{b^2} = 1$
 - eccentricity $e = v_1^2 - 1$
 - semi-major axis $a = \frac{1}{1 - e}$
 - semi-minor axis $b^2 = a^2(1 - e^2)$.

Parametric Elliptic and Hyperbolic Solutions (parametrised by θ)

- *Elliptic solutions*

- have $e < 1$ or $v_1 < \sqrt{2}$:
- $$\begin{aligned} x &= -ae + a \cos \theta, \\ y &= b' \sin \theta \quad (0 \leq \theta \leq 2\pi). \end{aligned}$$
- Semi-major axis a
- Semi-minor axis $b' = a\sqrt{1 - e^2}$.
- Focii $x_{\pm} = -ae \pm \sqrt{a^2 - b^2}$, so $x_{\pm} = 0, -2ae$
 - (the Sun is at $x_+ = 0$).

- *Hyperbolic solutions*

- have $e > 1$ or $v_1 > \sqrt{2}$:
- $$\begin{aligned} x &= -ae + a \cosh \theta \\ y &= b' \sinh \theta \quad (-\infty < \theta < \infty), \end{aligned}$$
- Semi-minor axis $b' = a\sqrt{e^2 - 1}$.