Ordinary Differential Equations

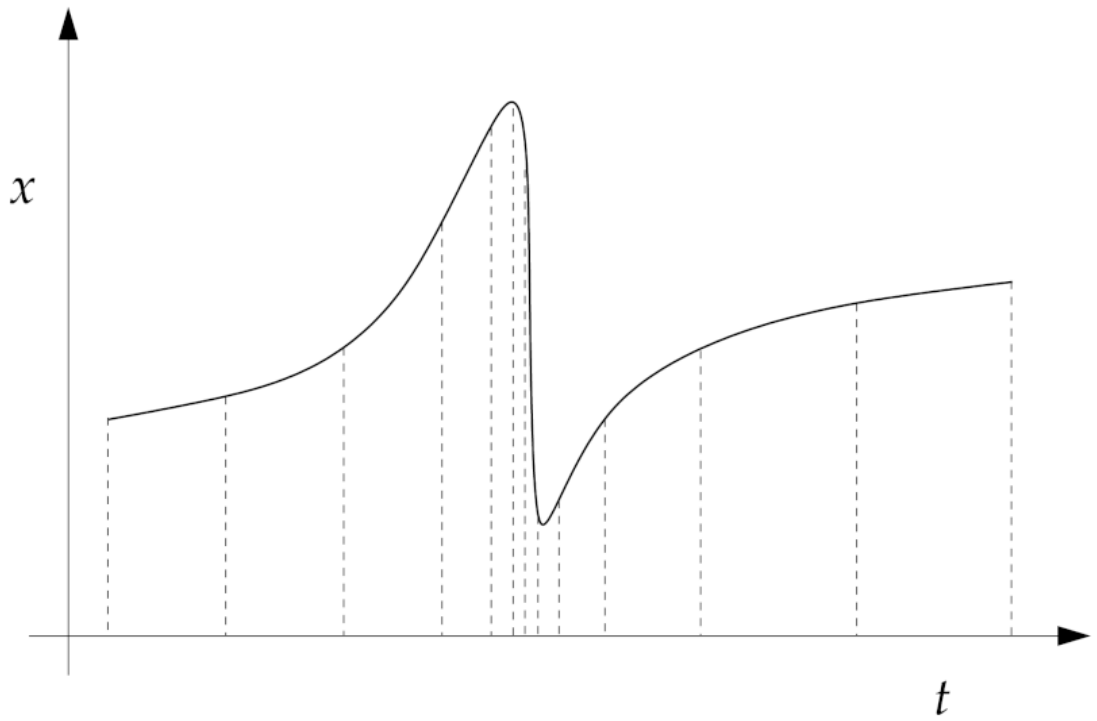
- Varying the step size.
- Verlet method.
- Leapfrog method.

Varying the step size

- In most cases, we can get better results if we allow the step size to vary during the running of the program, with the program choosing the best value at each step.

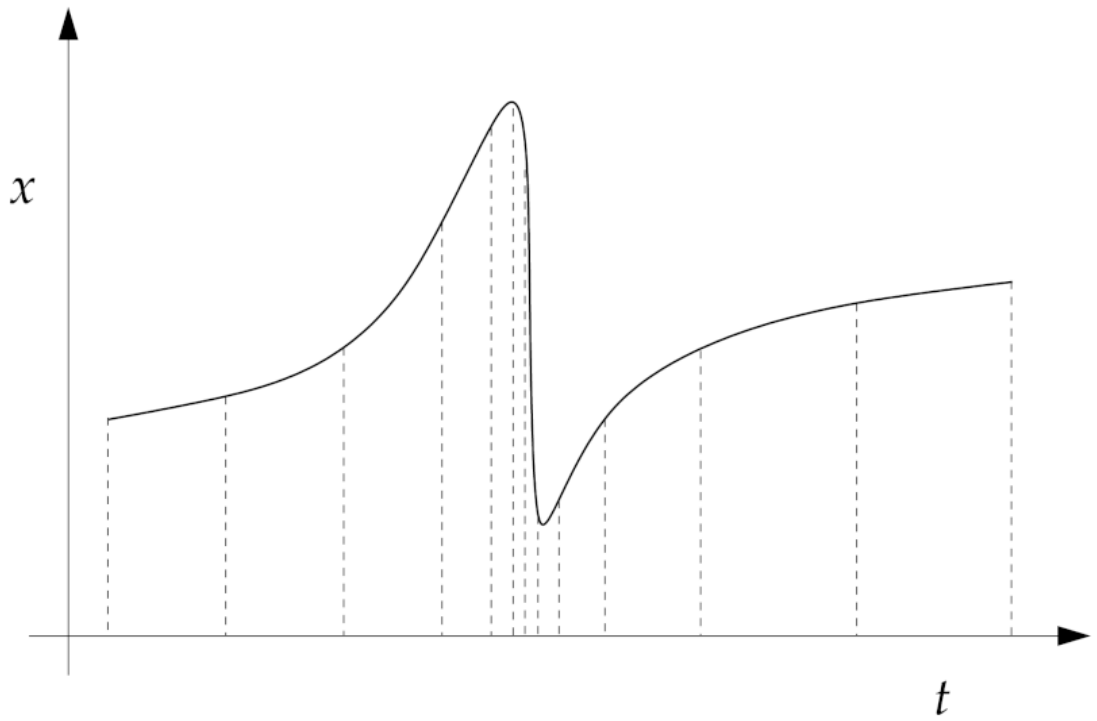
Varying the step size

- In most cases, we can get better results if we allow the step size to vary during the running of the program, with the program choosing the best value at each step.
- In the regions the solution is varying slowly, one needs few, widely spaced points.



Varying the step size

- In most cases, we can get better results if we allow the step size to vary during the running of the program, with the program choosing the best value at each step.
- In the regions the solution is varying slowly, one needs few, widely spaced points.
- In the regions the solution is varying rapidly, one need points that are more closely spaced!



Varying the step size

- Basic idea is to vary the step sizes h so that the error introduced per unit interval in t is roughly the same – $dx/dt = f(x, t)$.

Varying the step size

- Basic idea is to vary the step sizes h so that the error introduced per unit interval in t is roughly the same – $dx/dt = f(x, t)$.
- In practice:

Varying the step size

- Basic idea is to vary the step sizes h so that the error introduced per unit interval in t is roughly the same – $dx/dt = f(x, t)$.
- In practice:
  - First we have to estimate the error on our steps.

Varying the step size

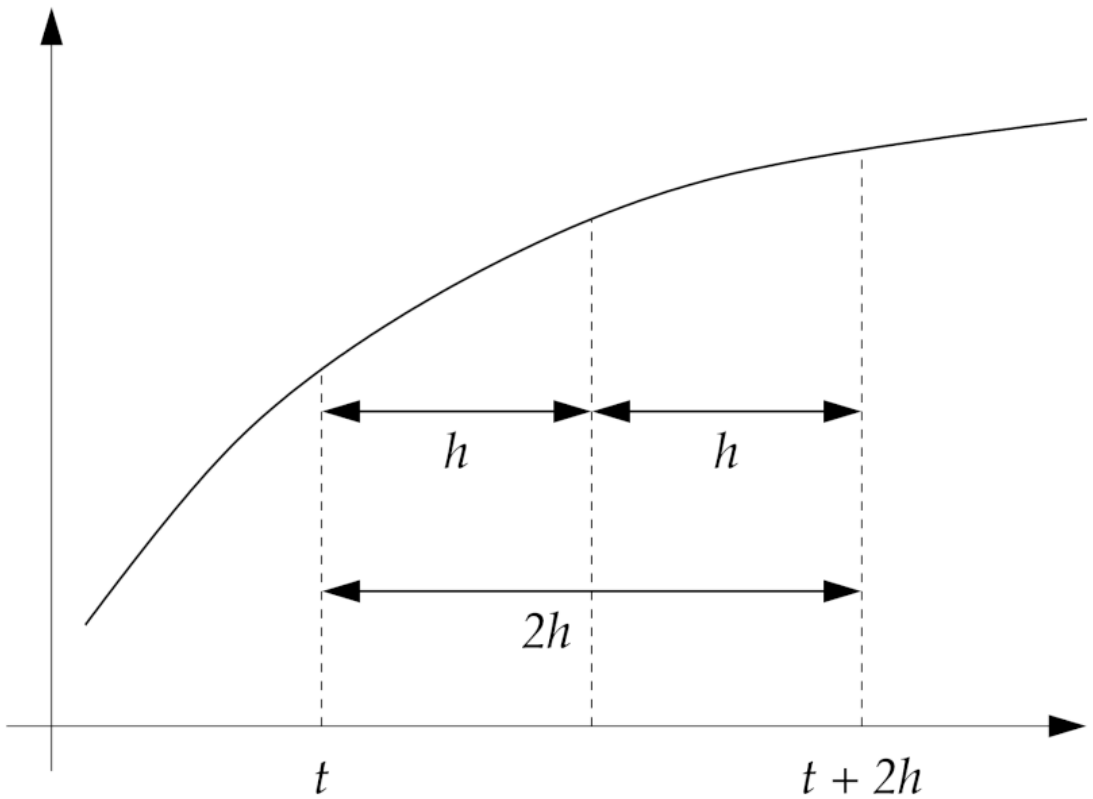
- Basic idea is to vary the step sizes h so that the error introduced per unit interval in t is roughly the same – $dx/dt = f(x, t)$.
- In practice:
  - First we have to estimate the error on our steps.
  - Then compare the error to required accuracy and either increase or decrease the step size to achieve the accuracy we want.

Varying the step size

- Basic idea is to vary the step sizes h so that the error introduced per unit interval in t is roughly the same – $dx/dt = f(x, t)$.
- In practice:
  - First we have to estimate the error on our steps.
  - Then compare the error to required accuracy and either increase or decrease the step size to achieve the accuracy we want.
- Illustrate this approach using 4th order Runge Kutta.

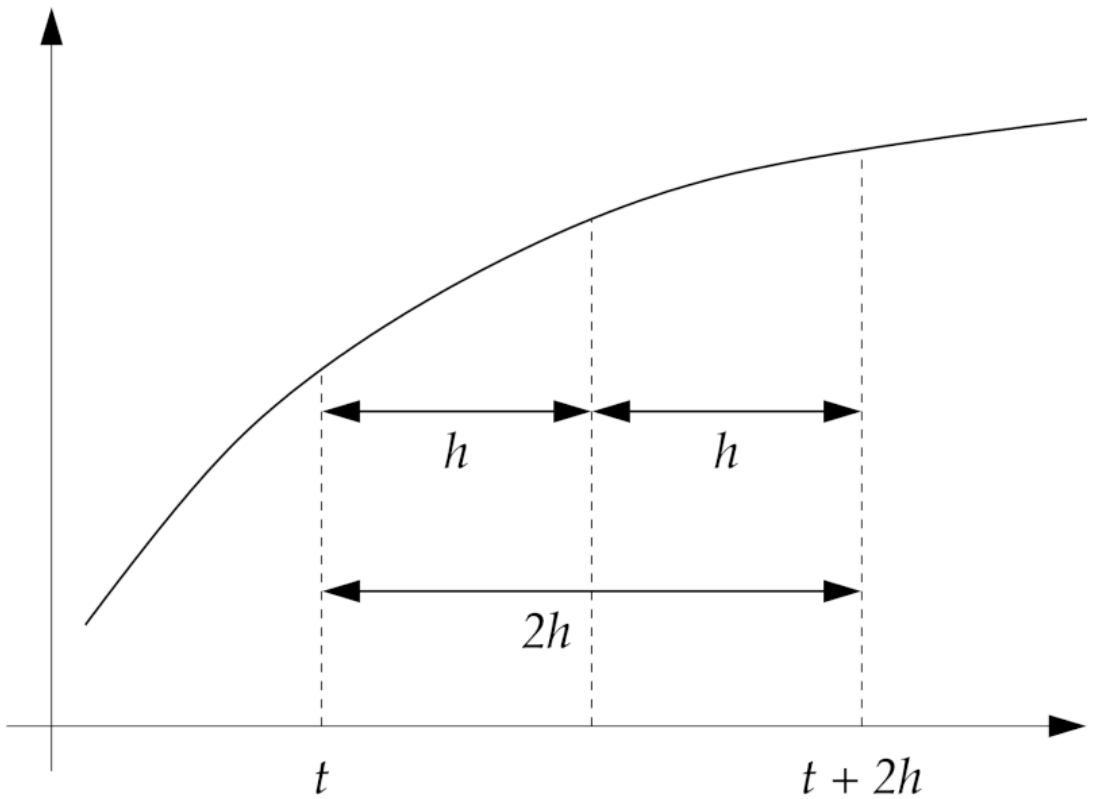
Varying the step size

- Start with some initial value of h – and do first two steps of the solution. So – starting at $x(t)$ we have an estimate for $x(t + 2h)$.



Varying the step size

- Start with some initial value of h – and do first two steps of the solution. So – starting at $x(t)$ we have an estimate for $x(t + 2h)$.
- Now going back to t again – we reach $x(t + 2h)$ in just one Runge-Kutta step.



Varying the step size

- Error in single step of 4th order Runge-Kutta is $\mathcal{O}(h^5)$.

Varying the step size

- Error in single step of 4th order Runge-Kutta is $\mathcal{O}(h^5)$.
- True value of $x(t+2h)$ from the first estimate:

$$x(t+2h) = x_1 + 2ch^5$$

Varying the step size

- Error in single step of 4th order Runge-Kutta is $\mathcal{O}(h^5)$.
- True value of $x(t+2h)$ from the first estimate:

$$x(t+2h) = x_1 + 2ch^5$$

- From the second $2h$ step:

$$x(t+2h) = x_2 + 32ch^5$$

Varying the step size

- Error in single step of 4th order Runge-Kutta is $\mathcal{O}(h^5)$.
- True value of $x(t+2h)$ from the first estimate:

$$x(t+2h) = x_1 + 2ch^5$$

- From the second $2h$ step:

$$x(t+2h) = x_2 + 32ch^5$$

- Equating the two, the error ϵ on steps of size h is:

$$\epsilon = ch^5 = \frac{1}{30}(x_1 - x_2)$$

Varying the step size

- Our goal is to make this error exactly equal to some target value.

Varying the step size

- Our goal is to make this error exactly equal to some target value.
- Let h' be the step size which makes this error reach the target value. Error in a single step is:

$$\begin{aligned}\epsilon' =& ch'^5 = ch^5\left(\frac{h'}{h}\right)^5 \\ =& \frac{1}{30}|x_1 - x_2|\left(\frac{h'}{h}\right)^5\end{aligned}$$

Varying the step size

- Our goal is to make this error exactly equal to some target value.
- Let h' be the step size which makes this error reach the target value. Error in a single step is:

$$\begin{aligned}\epsilon' =& ch'^5 = ch^5\left(\frac{h'}{h}\right)^5 \\ =& \tfrac{1}{30}|x_1 - x_2|\left(\frac{h'}{h}\right)^5\end{aligned}$$

- If our target accuracy per unit time is δ, the error for each step is $h'\delta$.

$$h' = h\left(\frac{30h\delta}{|x_1 - x_2|}\right)^{1/4} = h\rho^{1/4}$$

where $\rho = \frac{30h\delta}{|x_1 - x_2|}$.

Varying the step size

- Thus, after each step of the process (determining x_1 and x_2), depending on the value of ρ, the value of h in the next step is increased or decreased.

Varying the step size

- Thus, after each step of the process (determining x_1 and x_2), depending on the value of ρ, the value of h in the next step is increased or decreased.
- An upper bound is placed on the change in step size to prevent the problem when $x_1 \approx x_2$.

Varying the step size

- Thus, after each step of the process (determining x_1 and x_2), depending on the value of ρ, the value of h in the next step is increased or decreased.
- An upper bound is placed on the change in step size to prevent the problem when $x_1 \approx x_2$.
- If there are two variables, then one can define a composite error index, such as $\epsilon^2 = \epsilon_1^2 + \epsilon_2^2$ or just one of them depending on the situation.

Varying the step size

- Thus, after each step of the process (determining x_1 and x_2), depending on the value of ρ, the value of h in the next step is increased or decreased.
- An upper bound is placed on the change in step size to prevent the problem when $x_1 \approx x_2$.
- If there are two variables, then one can define a composite error index, such as $\epsilon^2 = \epsilon_1^2 + \epsilon_2^2$ or just one of them depending on the situation.
- This method is very widely used!

Local extrapolation

- Basically we know that:

$$x(t+2h) = x_1 + 2ch^5 + \mathcal{O}(h^6)$$

Local extrapolation

- Basically we know that:

$$x(t+2h) = x_1 + 2ch^5 + \mathcal{O}(h^6)$$

- Using the expression for ch^5:

$$x(t+2h) = x(t) + \tfrac{1}{15}(x_1 - x_2) + \mathcal{O}(h^6)$$

Local extrapolation

- Basically we know that:

$$x(t+2h) = x_1 + 2ch^5 + \mathcal{O}(h^6)$$

- Using the expression for ch^5:

$$x(t+2h) = x(t) + \tfrac{1}{15}(x_1 - x_2) + \mathcal{O}(h^6)$$

- This trick is called local extrapolation.

Local extrapolation

- Basically we know that:

$$x(t+2h) = x_1 + 2ch^5 + \mathcal{O}(h^6)$$

- Using the expression for ch^5:

$$x(t+2h) = x(t) + \tfrac{1}{15}(x_1 - x_2) + \mathcal{O}(h^6)$$

- This trick is called local extrapolation.
- This can be easily incorporated in the code and can give a modest improvement in the accuracy of the results.

The Leapfrog method

Consider:

$$\frac{dx}{dt} = f(x,t)$$

- In the second order Runge-Kutta method, one estimates the value of $x(t+h)$ by using the slope at the midpoint $f(x(t+\frac{1}{2}h), t+\frac{1}{2}h)$. But as one does not know the value at the midpoint, it is calculated using Euler's method:

$$\begin{aligned} x(t+\tfrac{1}{2}h) &= x(t) + \tfrac{1}{2}hf(x,t) \\ x(t+h) &= x(t) + hf(x(t+\tfrac{1}{2}h), t+\tfrac{1}{2}h) \end{aligned}$$

The Leapfrog method

Consider:

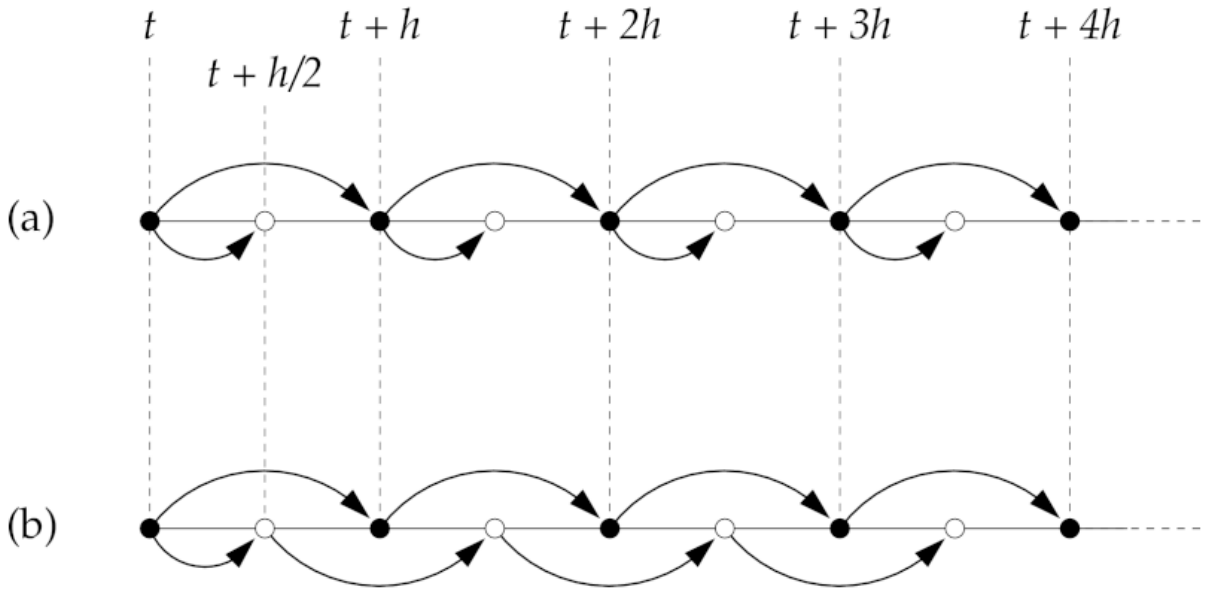
$$\frac{dx}{dt} = f(x,t)$$

- In the second order Runge-Kutta method, one estimates the value of $x(t+h)$ by using the slope at the midpoint $f(x(t+\frac{1}{2}h), t+\frac{1}{2}h)$. But as one does not know the value at the midpoint, it is calculated using Euler's method:

$$\begin{aligned} x(t+\tfrac{1}{2}h) &= x(t) + \tfrac{1}{2}hf(x,t) \\ x(t+h) &= x(t) + hf(x(t+\tfrac{1}{2}h), t+\tfrac{1}{2}h) \end{aligned}$$

- The leapfrog method is a variant of this idea – this method starts out the same way as Runge-Kutta with a half-step to the midpoint, followed by a full step to calculate $x(t+h)$ – but then rather than calculating the next midpoint value from $x(t+h)$, it is calculated from the previous midpoint value $x(t+\frac{1}{2}h)$.

The Leapfrog method



$$x(t+h) = x(t) + hf(x(t+\tfrac{1}{2}h), t+\tfrac{1}{2}h)$$
$$x(t+\tfrac{3}{2}h) = x(t+\tfrac{1}{2}h) + hf(x(t+h), t+h)$$

The Leapfrog method

- On the face of it, this method offers nothing more than a second-order Runge-Kutta method.

The Leapfrog method

- On the face of it, this method offers nothing more than a second-order Runge-Kutta method.
- It is time reversal symmetric, which makes it useful for physics problems where energy conservation is important.

The Leapfrog method

- On the face of it, this method offers nothing more than a second-order Runge-Kutta method.
- It is time reversal symmetric, which makes it useful for physics problems where energy conservation is important.
- The error is even in the step size h, which makes it an ideal starting point for Richardson extrapolation!

The Leapfrog method

- Suppose we start the leapfrog method by giving two values $x(t_1)$ and $x(t_1 + \frac{1}{2}h)$.

The Leapfrog method

- Suppose we start the leapfrog method by giving two values $x(t_1)$ and $x(t_1 + \frac{1}{2}h)$.
- Suppose we continue the solution to a later time $t = t_2$, calculating $x(t_2)$ and $x(t_2 + \frac{1}{2}h)$.

The Leapfrog method

- Suppose we start the leapfrog method by giving two values $x(t_1)$ and $x(t_1 + \frac{1}{2}h)$.
- Suppose we continue the solution to a later time $t = t_2$, calculating $x(t_2)$ and $x(t_2 + \frac{1}{2}h)$.
- Time reversal symmetry means that if we take these values and use the Leapfrog method backwards, with time interval $-h$, then we will retrace our steps and recover the values $x(t_1)$ and $x(t_1 + \frac{1}{2}h)$.

The Leapfrog method

- Suppose we start the leapfrog method by giving two values $x(t_1)$ and $x(t_1 + \frac{1}{2}h)$.
- Suppose we continue the solution to a later time $t = t_2$, calculating $x(t_2)$ and $x(t_2 + \frac{1}{2}h)$.
- Time reversal symmetry means that if we take these values and use the Leapfrog method backwards, with time interval $-h$, then we will retrace our steps and recover the values $x(t_1)$ and $x(t_1 + \frac{1}{2}h)$.
- This is important because it conserves energy.

The Leapfrog method

- Suppose we start the leapfrog method by giving two values $x(t_1)$ and $x(t_1 + \frac{1}{2}h)$.
- Suppose we continue the solution to a later time $t = t_2$, calculating $x(t_2)$ and $x(t_2 + \frac{1}{2}h)$.
- Time reversal symmetry means that if we take these values and use the Leapfrog method backwards, with time interval $-h$, then we will retrace our steps and recover the values $x(t_1)$ and $x(t_1 + \frac{1}{2}h)$.
- This is important because it conserves energy.
- Leapfrog method is time-reversal symmetric, but Runge-Kutta is not!!!

The Leapfrog method

- Suppose we start the leapfrog method by giving two values $x(t_1)$ and $x(t_1 + \frac{1}{2}h)$.
- Suppose we continue the solution to a later time $t = t_2$, calculating $x(t_2)$ and $x(t_2 + \frac{1}{2}h)$.
- Time reversal symmetry means that if we take these values and use the Leapfrog method backwards, with time interval $-h$, then we will retrace our steps and recover the values $x(t_1)$ and $x(t_1 + \frac{1}{2}h)$.
- This is important because it conserves energy.
- Leapfrog method is time-reversal symmetric, but Runge-Kutta is not!!!
- Does not mean that leapfrog gives exact solutions – the error is in third order.

The Leapfrog method

- Suppose we start the leapfrog method by giving two values $x(t_1)$ and $x(t_1 + \frac{1}{2}h)$.
- Suppose we continue the solution to a later time $t = t_2$, calculating $x(t_2)$ and $x(t_2 + \frac{1}{2}h)$.
- Time reversal symmetry means that if we take these values and use the Leapfrog method backwards, with time interval $-h$, then we will retrace our steps and recover the values $x(t_1)$ and $x(t_1 + \frac{1}{2}h)$.
- This is important because it conserves energy.
- Leapfrog method is time-reversal symmetric, but Runge-Kutta is not!!!
- Does not mean that leapfrog gives exact solutions – the error is in third order.
- Energy is conserved does not mean it is constant over some cycle – it may oscillate – but will not drift.

The Verlet method

- Suppose we are using the Leapfrog method to solve the classical equations of motion.

$$\frac{dx}{dt} = v \qquad \frac{dv}{dt} = f(x,t)$$

The Verlet method

- Suppose we are using the Leapfrog method to solve the classical equations of motion.

$$\frac{dx}{dt} = v \qquad \frac{dv}{dt} = f(x,t)$$

- Rather than using a Leapfrog method on the vector equation, $\mathbf{r} = (x, v)$, we will start by assuming that we are given x at some time t and the value of v at time $t + \frac{1}{2}h$:

$$\begin{aligned} x(t+h) &= x(t) + hv(t+\tfrac{1}{2}h) \\ v(t+\tfrac{3}{2}h) &= v(t+\tfrac{1}{2}h) + hf(x(t+h), t+h) \end{aligned}$$

The Verlet method

- Suppose we are using the Leapfrog method to solve the classical equations of motion.

$$\frac{dx}{dt} = v \qquad \frac{dv}{dt} = f(x,t)$$

- Rather than using a Leapfrog method on the vector equation, $\mathbf{r} = (x, v)$, we will start by assuming that we are given x at some time t and the value of v at time $t + \frac{1}{2}h$:

$$\begin{aligned} x(t+h) =& x(t) + hv(t+\tfrac{1}{2}h) \\ v(t+\tfrac{3}{2}h) =& v(t+\tfrac{1}{2}h) + hf(x(t+h), t+h) \end{aligned}$$

- The values of x are only available at integer multiples of h, while the velocity is at half integer multiples.

The Verlet Method

Given the initial values of x and v at some time t :

$$\begin{aligned} v(t+\tfrac{1}{2}h) &= v(t) + \tfrac{1}{2}hf(x(t),t) \\ x(t+h) &= c(t) + hv(t+\tfrac{1}{2}h) \\ k &= hf(x(t+h),t+h) \\ v(t+h) &= v(t+\tfrac{1}{2}h) + \tfrac{1}{2}k \\ v(t+\tfrac{3}{2}h) &= v(t+\tfrac{1}{2}h) + k \end{aligned}$$

Notice in the above equations, $v(t+h)$ is calculated even though it is not really needed. This is done to ensure that one can calculate total energy at any time, where both x and v are available.