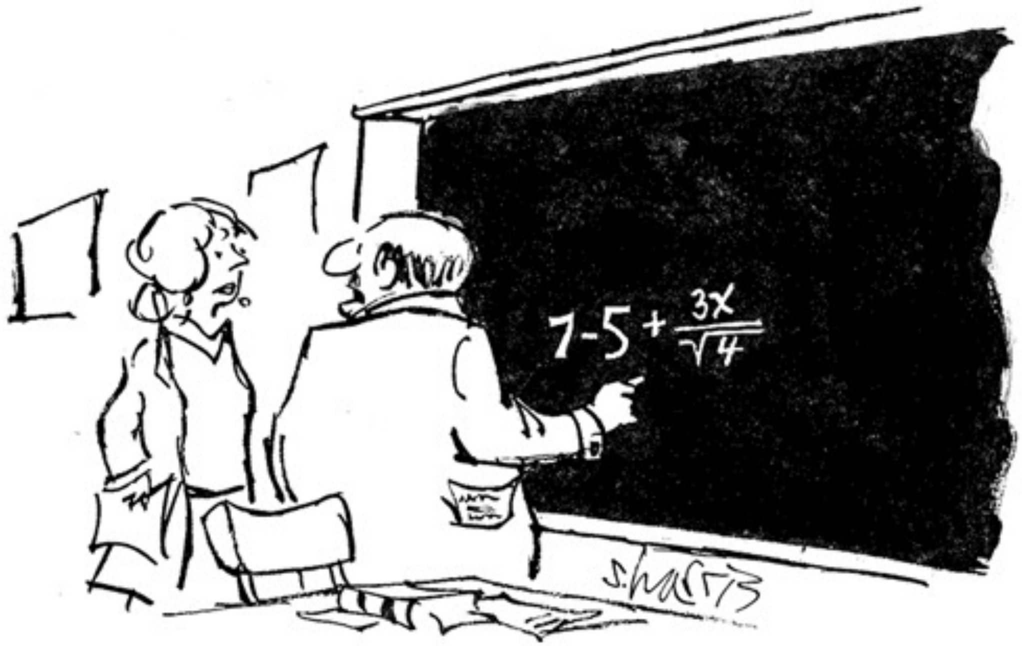


"YOU CAN'T IMAGINE HOW TIGHT OUR BUDGET IS. WE CAN ONLY WORK WITH SINGLE-DIGIT NUMBERS."

- the basics
- solving differential equations
- symplectic integration
- accuracy checks

- **the basics**
  - root finding
  - curve fitting
  - integration
- solving differential equations
- symplectic integration
- accuracy checks

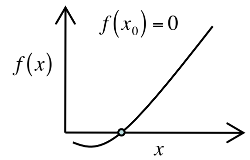
- **the basics**
  - *root finding*
  - curve fitting
  - integration
- solving differential equations
- symplectic integration
- accuracy checks

- root finding

Review of Numerical Methods

- root finding

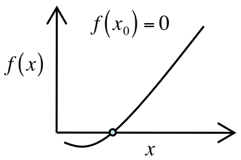
roots of functions



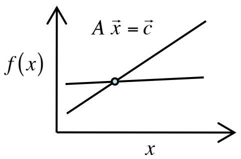
Review of Numerical Methods

- root finding

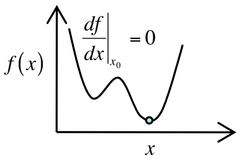
roots of functions



linear algebraic equations



optimisation



- root finding - example

$$m(z) = M + 5\log_{10}\left(\frac{c(1+z)}{\sqrt{|k|}}\ \gamma\left(\sqrt{|k|}\int_0^z \left[(1+z')^2(1+\Omega_m z') - z'(2+z')\Omega_\Lambda\right]^{-1/2} dz'\right)\right)$$

=> for a given SN-Ia explosion you aim to find it's redshift

- known: $m, M, \Omega_\Lambda, \Omega_m$
- unknown: z

- root finding - example

$$m(z) = M + 5\log_{10}\left(\frac{c(1+z)}{\sqrt{|k|}}\ \gamma\left(\sqrt{|k|}\int_0^z \left[(1+z')^2(1+\Omega_m z') - z'(2+z')\Omega_\Lambda\right]^{-1/2} dz'\right)\right)$$

=> for a given SN-Ia explosion you aim to find it's redshift

- known: $m, M, \Omega_\Lambda, \Omega_m$
- unknown: z

$$\Rightarrow \quad 0 = f(z) = m - \left(M + 5\log_{10}\left(\frac{c(1+z)}{\sqrt{|k|}}\ \gamma\left(\sqrt{|k|}\int_0^z \left[(1+z')^2(1+\Omega_m z') - z'(2+z')\Omega_\Lambda\right]^{-1/2} dz'\right)\right)\right)$$

- root finding - example

$$m(z) = M + 5\log_{10}\left(\frac{c(1+z)}{\sqrt{|k|}}\ \gamma\left(\sqrt{|k|}\int_0^z \left[(1+z')^2(1+\Omega_m z') - z'(2+z')\Omega_\Lambda\right]^{-1/2} dz'\right)\right)$$

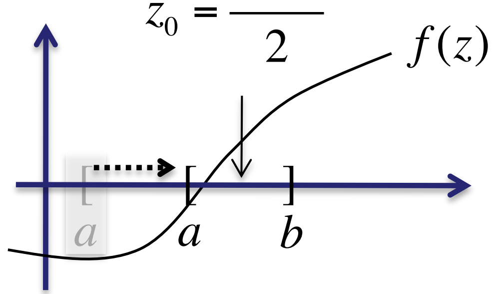
=> for a given SN-Ia explosion you aim to find it's redshift

- known: $m, M, \Omega_\Lambda, \Omega_m$
- unknown: z

$$\Rightarrow \quad 0 = f(z) = m - \left(M + 5\log_{10}\left(\frac{c(1+z)}{\sqrt{|k|}}\ \gamma\left(\sqrt{|k|}\int_0^z \left[(1+z')^2(1+\Omega_m z') - z'(2+z')\Omega_\Lambda\right]^{-1/2} dz'\right)\right)\right)$$

bi-section method:

$$z_0 = \frac{a+b}{2}$$



- root finding - example

$$m(z) = M + 5\log_{10}\left(\frac{c(1+z)}{\sqrt{|k|}}\ \gamma\left(\sqrt{|k|}\int_0^z\left[(1+z')^2(1+\Omega_m z') - z'(2+z')\Omega_\Lambda\right]^{-1/2}dz'\right)\right)$$

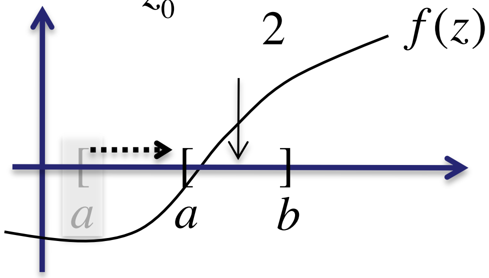
=> for a given SN-Ia explosion you aim to find it's redshift

- known: $m, M, \Omega_\Lambda, \Omega_m$
- unknown: z

$$\Rightarrow \quad 0 = f(z) = m - \left(M + 5\log_{10}\left(\frac{c(1+z)}{\sqrt{|k|}}\ \gamma\left(\sqrt{|k|}\int_0^z\left[(1+z')^2(1+\Omega_m z') - z'(2+z')\Omega_\Lambda\right]^{-1/2}dz'\right)\right)\right)$$

bi-section method:

$$z_0 = \frac{a+b}{2}$$

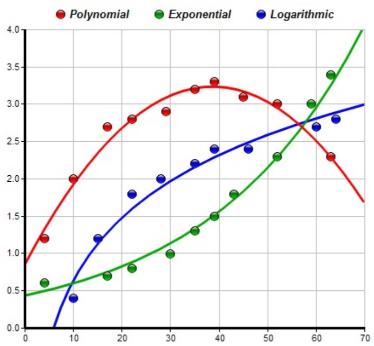



```
while(abs(b-a)>eps) {
   z0 = (a+b)/2;
   if(f(a)*f(z0) > 0)
       a = z0;
   else
       b = z0;
}
```

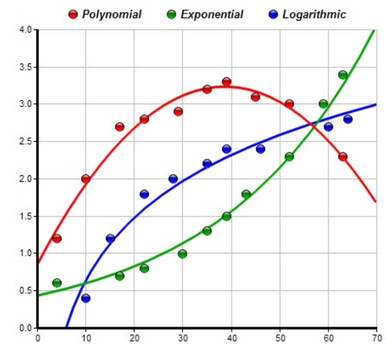
- **the basics**
  - root finding
  - *curve fitting*
  - integration
- solving differential equations
- symplectic integration
- accuracy checks

Review of Numerical Methods

- curve fitting



Review of Numerical Methods



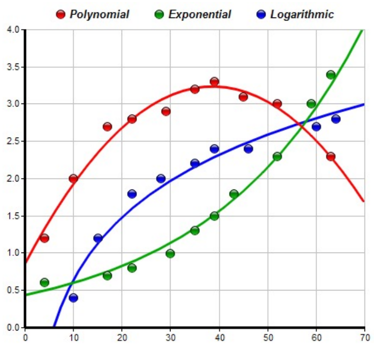
- curve fitting

  - **linear least-square fitting**

    - in case $f(x,a_1,a_2,a_3,...)$ only contains a linear combination of the N_{param} fit parameters $a_1,a_2,a_3,...$

$$f(x,\vec{a}) = \sum_{j=1}^{N_{param}} a_j \, \phi_j(x)$$

we can find a solution to the least-square minimization by solving a linear system for $\boldsymbol{a}=(a_1,a_2,a_3,...)$



- curve fitting

  - **linear least-square fitting**

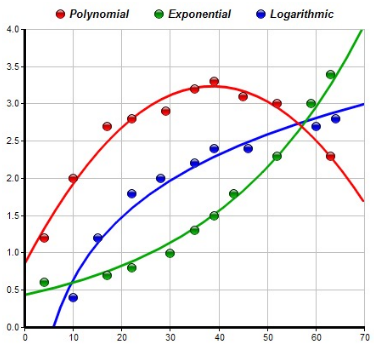
    - in case $f(x,a_1,a_2,a_3,\ldots)$ only contains a linear combination of the N_{param} fit parameters $a_1,a_2,a_3,\ldots$

$$f(x,\vec{a}) = \sum_{j=1}^{N_{param}} a_j\ \phi_j(x)$$

we can find a solution to the least-square minimization by solving a linear system for $\boldsymbol{a}=(a_1,a_2,a_3,\ldots)$

- the optimization is then equivalent to solving

$$0 = \sum_{i=1}^{N} \left(y_i - f(x_i,\vec{a})\right) \phi_j(x_i) \qquad \forall j \in N_{param}$$



• curve fitting

▪ linear least-square fitting

- in case $f(x,a_1,a_2,a_3,...)$ only contains a linear combination of the N_{param} fit parameters $a_1,a_2,a_3,...$

$$f(x,\vec{a}) = \sum_{j=1}^{N_{param}} a_j \, \phi_j(x)$$

we can find a solution to the least-square minimization by solving a linear system for $\boldsymbol{a}=(a_1,a_2,a_3,...)$

- the optimization is then equivalent to solving

$$0 = \sum_{i=1}^{N} \left(y_i - f(x_i,\vec{a})\right) \phi_j(x_i) \qquad \forall j \in N_{param}$$

which is a linear system for $a_1,a_2,a_3,...$

$$\begin{pmatrix} a_1 \\ \ldots \\ a_{N_{param}} \end{pmatrix} = \left(X^T X\right)^{-1} X^T \begin{pmatrix} y_1 \\ \ldots \\ y_N \end{pmatrix} \qquad X_{ij} = \frac{\partial f(x_i,\vec{a})}{\partial a_j} = \phi_j(x_i) \Rightarrow X = \begin{pmatrix} \phi_1(x_1) & \ldots & \phi_{Nparam}(x_1) \\ \ldots & \ldots & \ldots \\ \phi_1(x_N) & \ldots & \phi_{Nparam}(x_N) \end{pmatrix}$$

i: line index
j: column index

$(x_1,y_1),\ldots,(x_N,y_N)$ are your data points

- **Note:** the $\phi_j(x)$ can be arbitrary (non-linear!) functions of x

- **the basics**
  - root finding
  - curve fitting
  - *integration*
- solving differential equations
- symplectic integration
- accuracy checks

- we want to calculate the definite integral...

$$I = \int_a^b f(x)dx$$

- we want to calculate the definite integral...

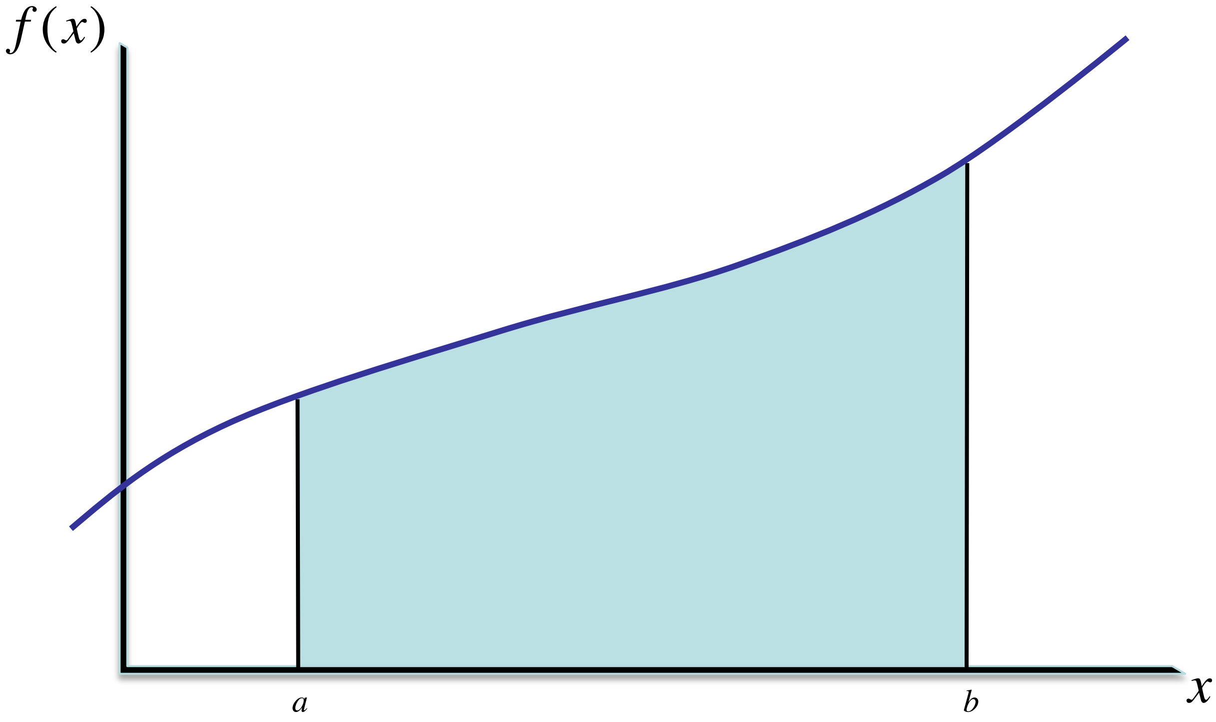
$$I = \int_a^b f(x)dx$$

- no difference to solving for the anti-derivative, i.e. let $b = x$...

$$F(x) = \int_{x_0}^{\boxed{x}} f(x)dx$$

- we want to calculate the definite integral...

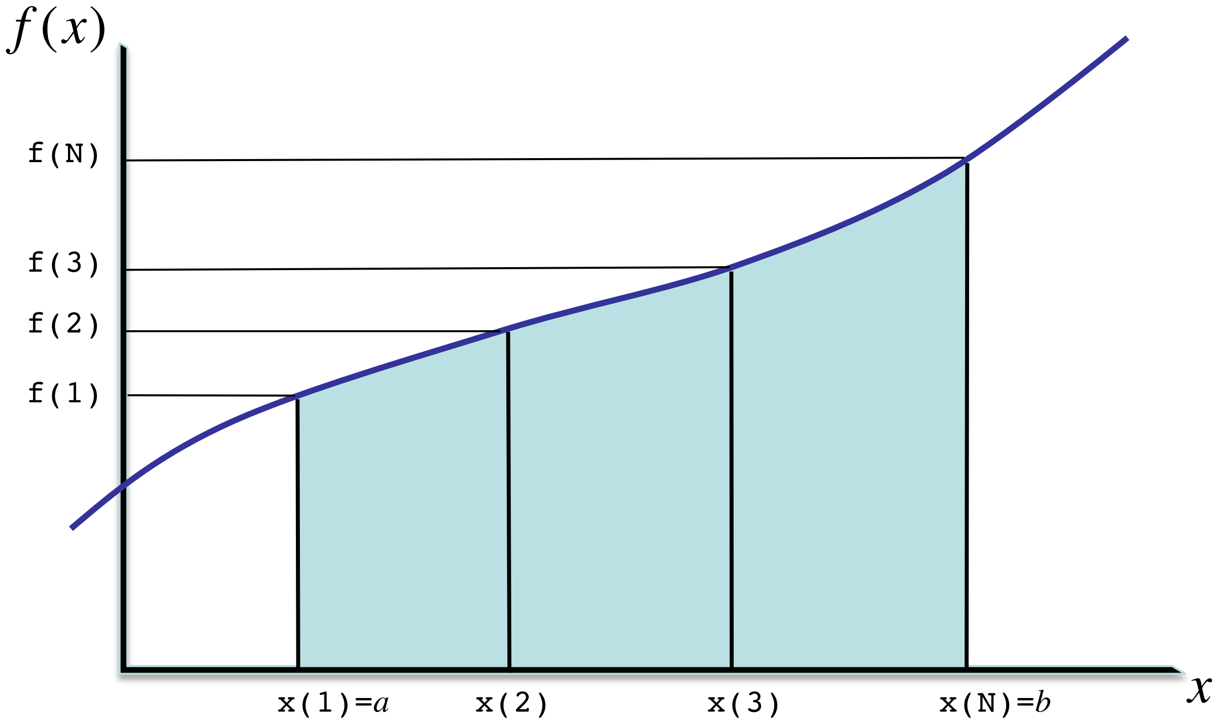
$$I = \int_a^b f(x)dx$$



- we want to calculate the definite integral...

$$I = \int_a^b f(x)dx$$

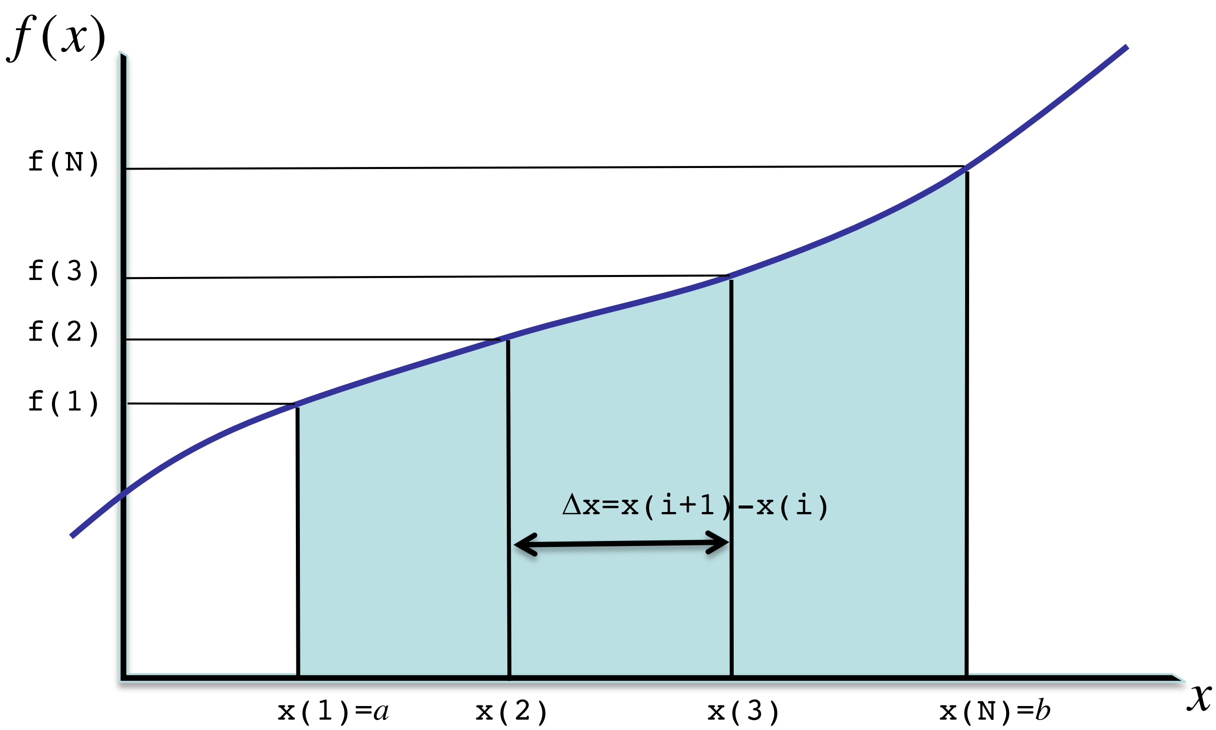
...given two arrays `x( )` and `f( )`



- we want to calculate the definite integral...

$$I = \int_a^b f(x)dx$$

...given two arrays `x()` and `f()`

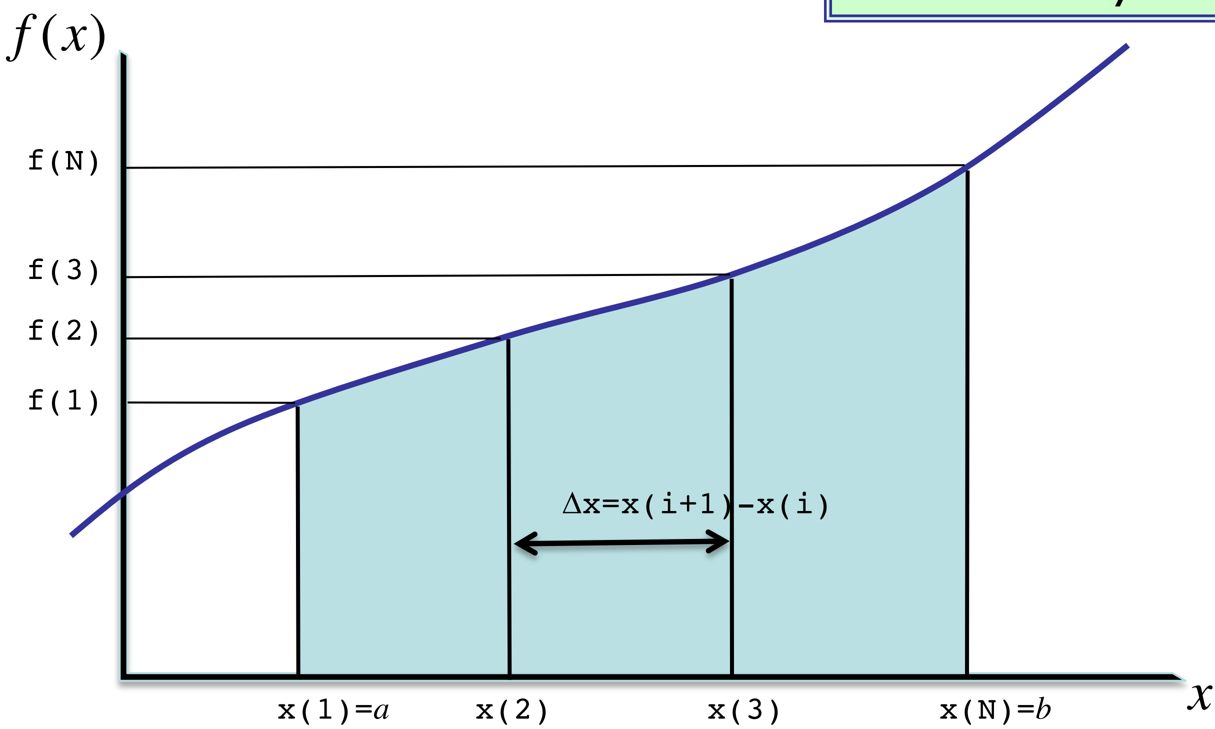


- we want to calculate the definite integral...

$$I = \int_a^b f(x)dx$$

...given two arrays `x()` and `f()`

we have several options to numerically "estimate" the integral

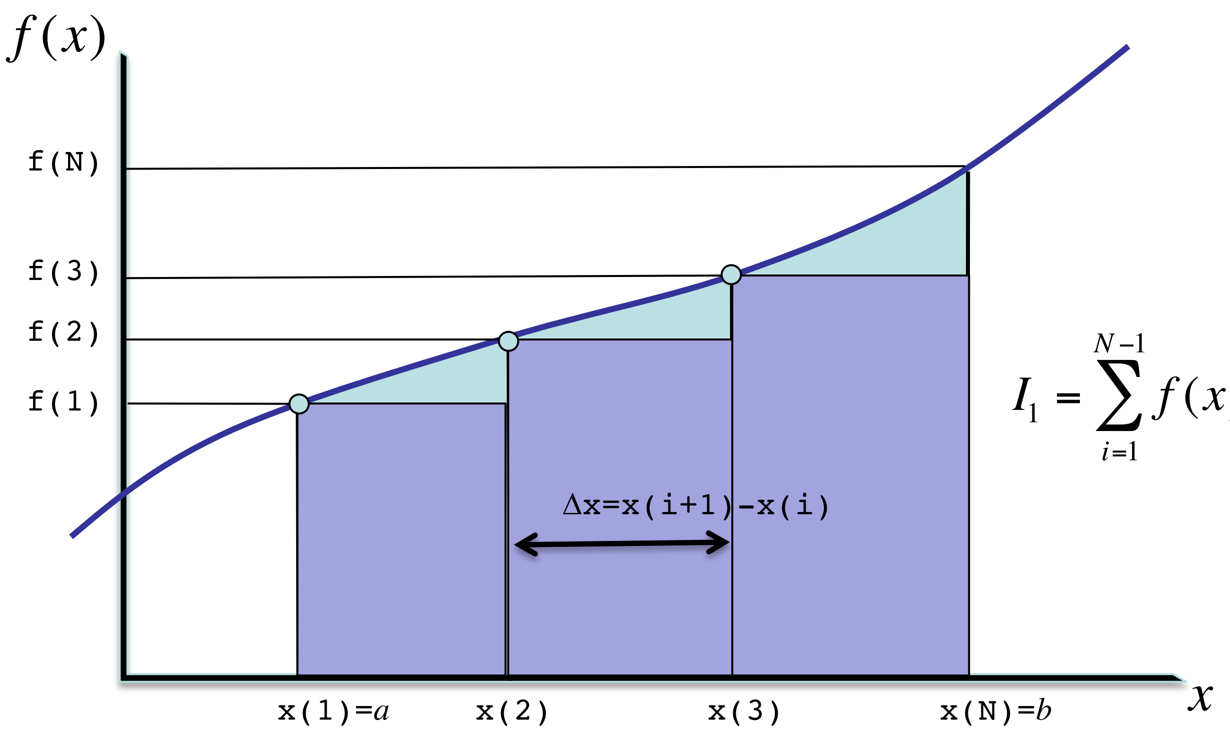


- we want to calculate the definite integral...

$$I = \int_a^b f(x)dx$$

...given two arrays `x()` and `f()`

option #1



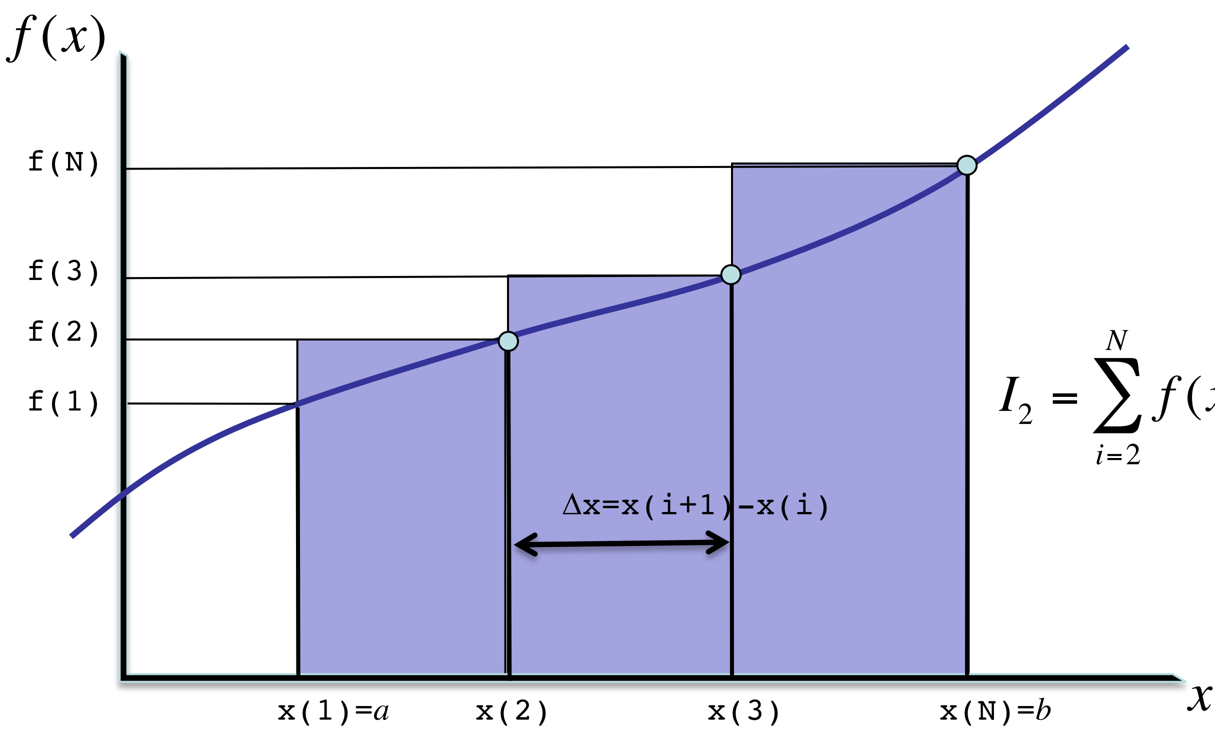
$$I_1 = \sum_{i=1}^{N-1} f(x_i)\,(x_{i+1} - x_i)$$

- we want to calculate the definite integral...

$$I = \int_a^b f(x)dx$$

...given two arrays `x( )` and `f( )`

option #2



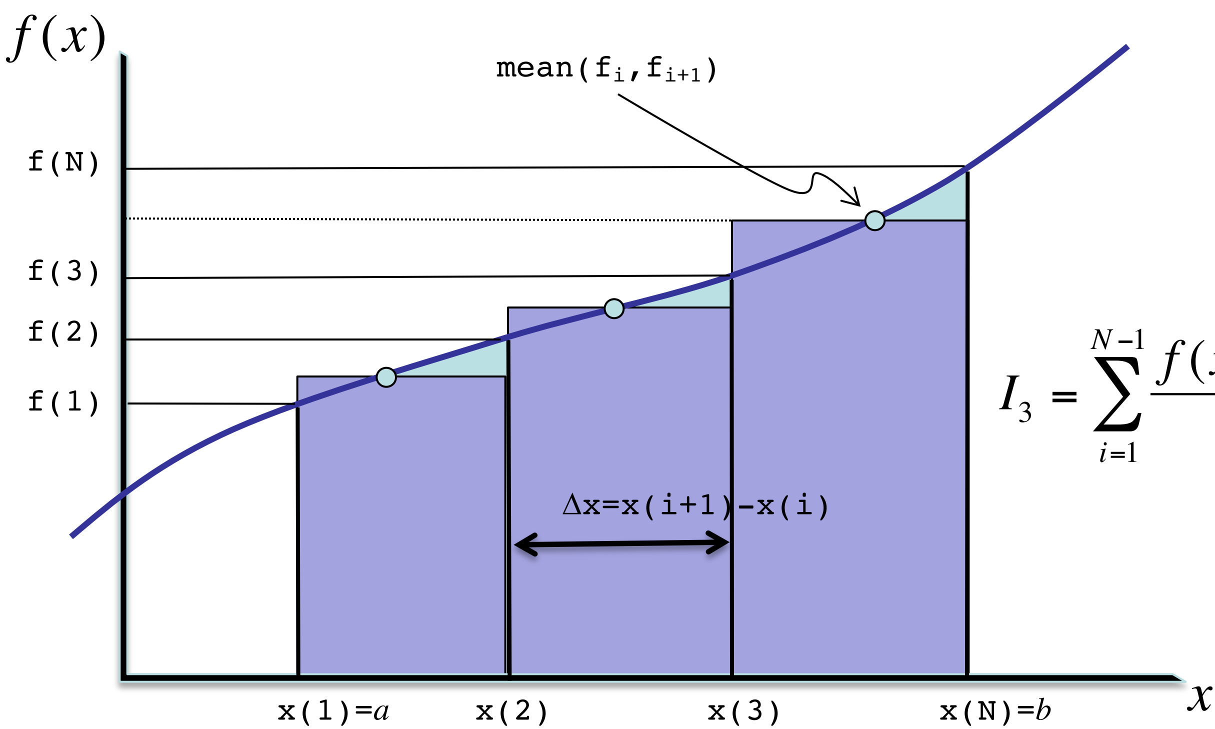
$$I_2 = \sum_{i=2}^{N} f(x_i)\,(x_i - x_{i-1})$$

- we want to calculate the definite integral...

$$I = \int_a^b f(x)dx$$

...given two arrays `x( )` and `f( )`

option #3



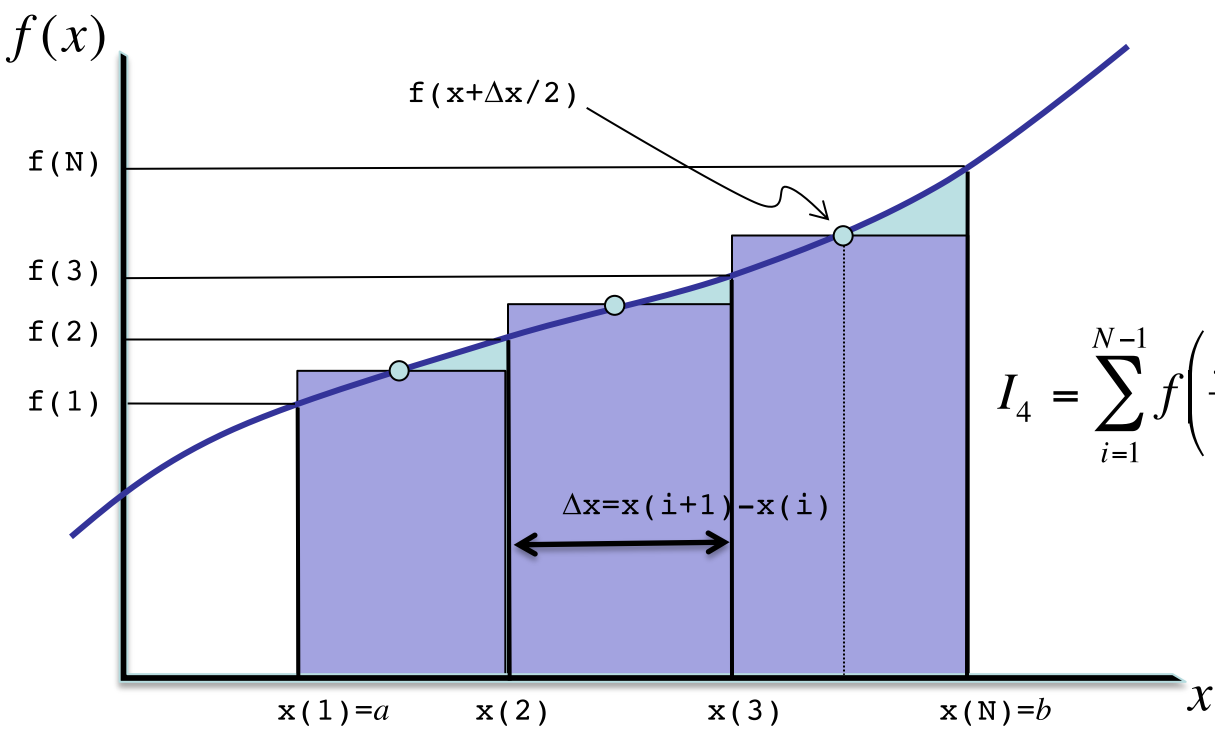
$$I_3 = \sum_{i=1}^{N-1} \frac{f(x_{i+1}) + f(x_i)}{2} (x_{i+1} - x_i)$$

- we want to calculate the definite integral...

$$I = \int_a^b f(x)dx$$

...given two arrays `x()` and `f()`

option #4



$$I_4 = \sum_{i=1}^{N-1} f\left(\frac{x_{i+1}+x_i}{2}\right)(x_{i+1}-x_i)$$

- we want to calculate the definite integral...

$$I = \int_a^b f(x)dx$$

...given two arrays `x()` and `f()`

- option 1 $$I_1 = \sum_{i=1}^{N-1} f(x_i)\,(x_{i+1} - x_i)$$

- option 2 $$I_2 = \sum_{i=2}^{N} f(x_i)\,(x_i - x_{i-1})$$

- option 3 $$I_3 = \sum_{i=1}^{N-1} \frac{f(x_{i+1}) + f(x_i)}{2}\,(x_{i+1} - x_i)$$

- option 4 $$I_4 = \sum_{i=1}^{N-1} f\left(\frac{x_{i+1} + x_i}{2}\right)(x_{i+1} - x_i)$$

- we want to calculate the definite integral...

$$I = \int_a^b f(x)dx$$

...given two arrays `x()` and `f()`

- option 1 $$I_1 = \sum_{i=1}^{N-1} f(x_i)\,(x_{i+1} - x_i)$$

- option 2 $$I_2 = \sum_{i=2}^{N} f(x_i)\,(x_i - x_{i-1})$$

- option 3 $$I_3 = \sum_{i=1}^{N-1} \frac{f(x_{i+1}) + f(x_i)}{2}\,(x_{i+1} - x_i)$$

- option 4 $$I_4 = \sum_{i=1}^{N-1} f\left(\frac{x_{i+1} + x_i}{2}\right)(x_{i+1} - x_i)$$

equal to 1st order

- we want to calculate the definite integral...

$$I = \int_a^b f(x)dx$$

...given two arrays `x( )` and `f( )`

- option 1 $$I_1 = \sum_{i=1}^{N-1} f(x_i)\,(x_{i+1} - x_i)$$

- option 2 $$I_2 = \sum_{i=2}^{N} f(x_i)\,(x_i - x_{i-1})$$

- option 3 $$I_3 = \sum_{i=1}^{N-1} \frac{f(x_{i+1}) + f(x_i)}{2}\,(x_{i+1} - x_i)$$

- option 4 $$I_4 = \sum_{i=1}^{N-1} f\left(\frac{x_{i+1} + x_i}{2}\right)(x_{i+1} - x_i)$$

$$\begin{aligned}
f\left(\frac{x_{i+1}+x_i}{2}\right) &= f\left(x_i + \frac{x_{i+1}-x_i}{2}\right) \\
&= f(x_i) + f'(x_i)\left(x_i + \frac{x_{i+1}-x_i}{2} - x_i\right) + \ldots \\
\text{1}^{\text{st}}\text{ order} \quad &\approx f(x_i) + f'(x_i)\left(\frac{x_{i+1}-x_i}{2}\right) \\
&= f(x_i) + \frac{1}{2} f'(x_i)(x_{i+1}-x_i) \\
\text{1}^{\text{st}}\text{ order} \quad &\approx f(x_i) + \frac{1}{2}\frac{f(x_{i+1}) - f(x)}{(x_{i+1}-x_i)}(x_{i+1}-x_i) \\
&= \frac{2f(x_i) + f(x_{i+1}) - f(x)}{2} \\
&= \frac{f(x_i) + f(x_{i+1})}{2}
\end{aligned}$$

Review of Numerical Methods

- the basics
- **solving differential equations**
- symplectic integration
- accuracy checks

- actual set of equations *(coupled set of first order partial differential equations)*

  - collisionless matter (e.g. dark matter)

$$\frac{d\vec{x}_{DM}}{dt} = \vec{v}_{DM}$$
$$\frac{d\vec{v}_{DM}}{dt} = -\nabla\phi$$

  - collisional matter (e.g. gas)

$$\Delta\phi = 4\pi G\rho_{tot}$$

$$\frac{\partial\rho}{\partial t} + \nabla\cdot\left(\rho\vec{v}\right) = 0$$

$$\frac{\partial\left(\rho\vec{v}\right)}{\partial t} + \nabla\cdot\left(\rho\vec{v}\otimes\vec{v} + \left(p + \frac{1}{2\mu}B^2\right)\vec{1} - \frac{1}{\mu}\vec{B}\otimes\vec{B}\right) = \rho\ \left(-\nabla\phi\right)$$

$$\frac{\partial\left(\rho E\right)}{\partial t} + \nabla\cdot\left(\left[\rho E + p + \frac{1}{2\mu}B^2\right]\vec{v} - \frac{1}{\mu}\left[\vec{v}\cdot\vec{B}\right]\vec{B}\right) = \rho\vec{v}\cdot\left(-\nabla\phi\right) + \left(\Gamma - L\right)$$

$$\frac{\partial\vec{B}}{\partial t} + \nabla\times\left(-\vec{v}\times\vec{B}\right) = 0$$

$$p = \left(\gamma - 1\right)\rho\varepsilon$$

$$\rho\varepsilon = \rho E - \frac{1}{2}\rho v^2$$

- actual set of equations (*coupled set of first order partial differential equations*)

  - collisionless matter (e.g. dark matter)

$$\frac{d\vec{x}_{DM}}{dt} = \vec{v}_{DM}$$
$$\frac{d\vec{v}_{DM}}{dt} = -\nabla\phi$$

  - collisional matter (e.g. gas)

$$\Delta\phi = 4\pi G\rho_{tot}$$

coupling via Poisson's equation

$$\frac{\partial\rho}{\partial t} + \nabla\cdot\left(\rho\vec{v}\right) = 0$$
$$\frac{\partial(\rho\vec{v})}{\partial t} + \nabla\cdot\left(\rho\vec{v}\otimes\vec{v} + \left(p + \frac{1}{2\mu}B^2\right)\vec{1} - \frac{1}{\mu}\vec{B}\otimes\vec{B}\right) = \rho\left(-\nabla\phi\right)$$
$$\frac{\partial(\rho E)}{\partial t} + \nabla\cdot\left(\left[\rho E + p + \frac{1}{2\mu}B^2\right]\vec{v} - \frac{1}{\mu}\left[\vec{v}\cdot\vec{B}\right]\vec{B}\right) = \rho\vec{v}\cdot\left(-\nabla\phi\right) + \left(\Gamma - L\right)$$
$$\frac{\partial\vec{B}}{\partial t} + \nabla\times\left(-\vec{v}\times\vec{B}\right) = 0$$

$$p = (\gamma - 1)\rho\varepsilon$$
$$\rho\varepsilon = \rho E - \frac{1}{2}\rho v^2$$

- actual set of equations *(coupled set of first order partial differential equations)*
  - collisionless matter (e.g. dark matter)

$$\frac{d\vec{x}_{DM}}{dt} = \vec{v}_{DM}$$

$$\frac{d\vec{v}_{DM}}{dt} = -\nabla\phi$$

  - collisional matter (e.g. gas)

$$\Delta\phi = 4\pi G\rho_{tot}$$

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\vec{v}) = 0$$

$$\frac{\partial(\rho\vec{v})}{\partial t} + \nabla\cdot\left(\rho\vec{v}\otimes\vec{v} + \left(p + \frac{1}{2\mu}B^2\right)\vec{1} - \frac{1}{\mu}\vec{B}\otimes\vec{B}\right) = \rho\,(-\nabla\phi)$$

$$\frac{\partial(\rho E)}{\partial t} + \nabla\cdot\left(\left[\rho E + p + \frac{1}{2\mu}B^2\right]\vec{v} - \frac{1}{\mu}\left[\vec{v}\cdot\vec{B}\right]\vec{B}\right) = \rho\vec{v}\cdot(-\nabla\phi) + (\Gamma - L)$$

$$\frac{\partial\vec{B}}{\partial t} + \nabla\times(-\vec{v}\times\vec{B}) = 0$$

$$p = (\gamma - 1)\rho\varepsilon$$

$$\rho\varepsilon = \rho E - \frac{1}{2}\rho v^2$$

Review of Numerical Methods

- ordinary differential equation

$$0 = G(f^{(n)}, f^{(n-1)}, \ldots, f^{(2)}, f^{(1)}, f^{(0)}, t) \qquad f^{(n)} \equiv \frac{d^n f}{dt^n}$$

Review of Numerical Methods

- 1st order ordinary differential equation

$$0 = G(\frac{df}{dt}, f, t)$$

Review of Numerical Methods

- 1^{st} order explicit ordinary differential equation

$$\frac{df}{dt} = G(f, t)$$

- 1st order explicit ordinary differential equation

$$\frac{df}{dt} = G(f,t)$$

$$\left.\begin{aligned}
\frac{d\vec{x}_{DM}}{dt} &= \vec{v}_{DM} \\
\frac{d\vec{v}_{DM}}{dt} &= -\nabla\phi \\
\Delta\phi &= 4\pi G\rho_{tot} \\
\frac{\partial\rho}{\partial t} + \nabla\cdot\left(\rho\vec{v}\right) &= 0 \\
\frac{\partial\left(\rho\vec{v}\right)}{\partial t} + \nabla\cdot\left(\rho\vec{v}\otimes\vec{v} + \left(p + \frac{1}{2\mu}B^2\right)\vec{1} - \frac{1}{\mu}\vec{B}\otimes\vec{B}\right) &= \rho\;\left(-\nabla\phi\right) \\
\frac{\partial\left(\rho E\right)}{\partial t} + \nabla\cdot\left(\left[\rho E + p + \frac{1}{2\mu}B^2\right]\vec{v} - \frac{1}{\mu}\left[\vec{v}\cdot\vec{B}\right]\vec{B}\right) &= \rho\vec{v}\cdot\left(-\nabla\phi\right) + \left(\Gamma - L\right) \\
\frac{\partial\vec{B}}{\partial t} + \nabla\times\left(-\vec{v}\times\vec{B}\right) &= 0
\end{aligned}\right\}\text{explicit}$$

Review of Numerical Methods

- 1st order explicit ordinary differential equation

$$\frac{df}{dt} = G(f, t)$$

`f()` and `t()` are given in two arrays, i.e. f and t are **not** continuous

Review of Numerical Methods

- 1[st] order explicit ordinary differential equation

$$\frac{df}{dt} = G(f,t)$$

$$=> \quad \frac{\Delta f}{\Delta t} = \frac{f(t_{i+1}) - f(t_i)}{t_{i+1} - t_i} = \frac{f_{i+1} - f_i}{t_{i+1} - t_i} = G(f_i, t_i)$$

$$=> \quad f_{i+1} = f_i + \Delta t \, G(f_i, t_i)$$

Review of Numerical Methods

- 1^{st} order explicit ordinary differential equation

$$\frac{df}{dt} = G(f,t)$$

$$=> \quad \frac{\Delta f}{\Delta t} = \frac{f(t_{i+1}) - f(t_i)}{t_{i+1} - t_i} = \frac{f_{i+1} - f_i}{t_{i+1} - t_i} = G(f_i, t_i)$$

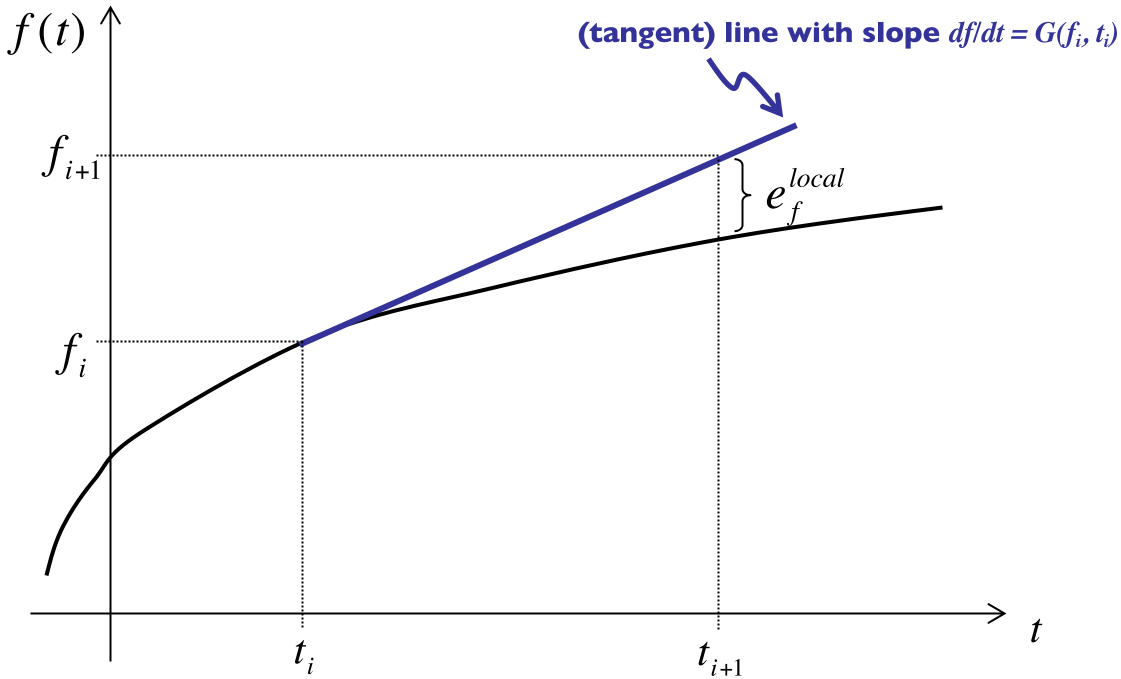
$$=> \quad f_{i+1} = f_i + \Delta t\, G(f_i, t_i)$$

Taylor expansion of $f(t)$ about t_i up to 1^{st} order...

$$f(t_i + \Delta t) = f(t_i) + \left.\frac{df}{dt}\right|_{t_i} dt$$

- 1st order explicit ordinary differential equation $$\frac{df}{dt} = G(f,t)$$

  - Euler scheme



$$\Rightarrow \quad f_{i+1} = f_i + \Delta t \, G(f_i, t_i)$$

Review of Numerical Methods

- 1[st] order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$
    - Euler scheme

$$f_{i+1} = f_i + \Delta t \, G(f_i, t_i)$$

Review of Numerical Methods

- 1st order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$

  - Euler scheme

$$f_{i+1} = f_i + \Delta t\, G(f_i, t_i)$$

  - local error estimate (F = correct solution, f = numerical solution)

$$F(t_{i+1}) = F(t_i) + \Delta t\, \dot{F}(t_i) + \frac{(\Delta t)^2}{2}\ddot{F}(t_i) + \ldots$$

$$e_f^{local} = F(t_{i+1}) - f_{i+1} = F(t_i) + \Delta t\, \dot{F}(t_i) + \frac{(\Delta t)^2}{2}\ddot{F}(t_i) + \ldots - \left(f_i + \Delta t\, G(f_i, t_i)\right)$$

=> $e_f^{local} \propto (\Delta t)^2$

- 1[st] order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$

  - Euler scheme

$$f_{i+1} = f_i + \Delta t\; G(f_i, t_i)$$

  - global error estimate

$$e_f^{global} \propto \sum_{i=1}^{N} (\Delta t)^2 = N(\Delta t)^2 = \frac{T_N - T_0}{\Delta t}(\Delta t)^2 \propto \Delta t$$

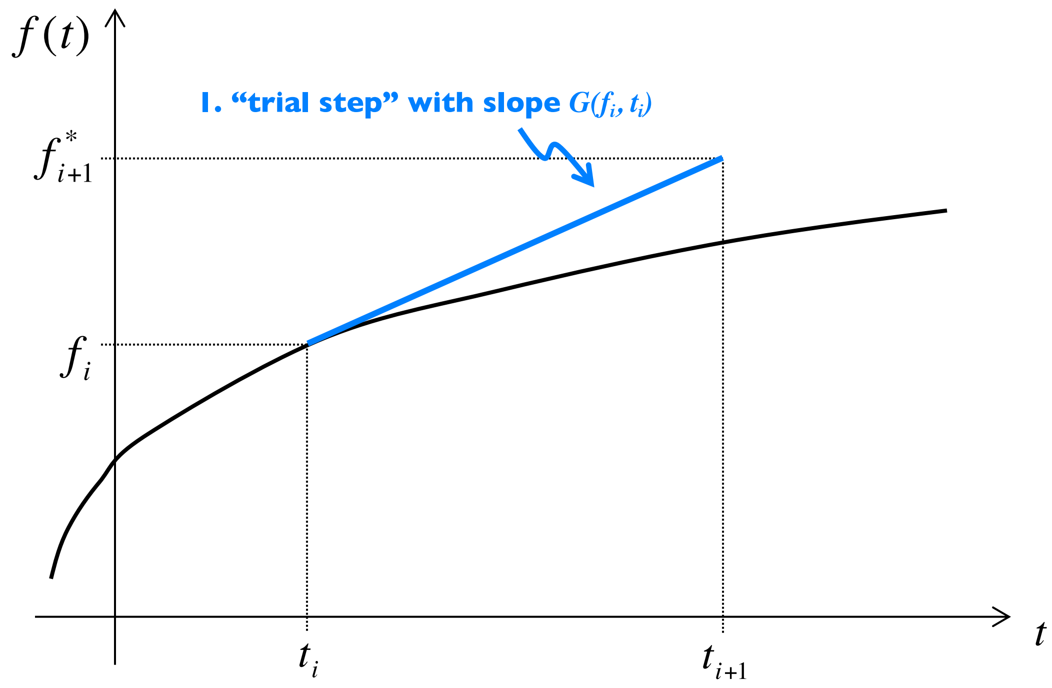
=> $e_f^{global} \propto (\Delta t)$ "first order accurate"

increase accuracy by including higher derivates…

Review of Numerical Methods

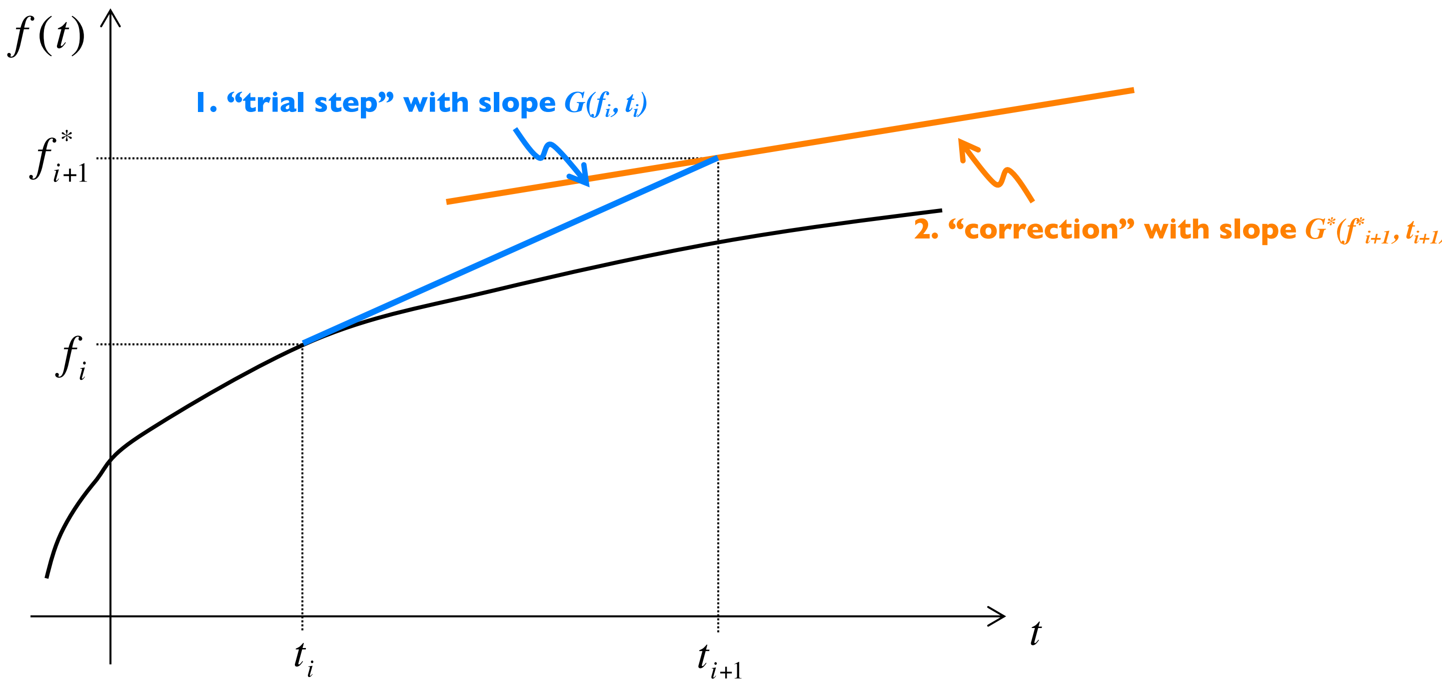
- 1st order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$

  - modified Euler scheme



- 1st order explicit ordinary differential equation $\dfrac{df}{dt} = G(f,t)$

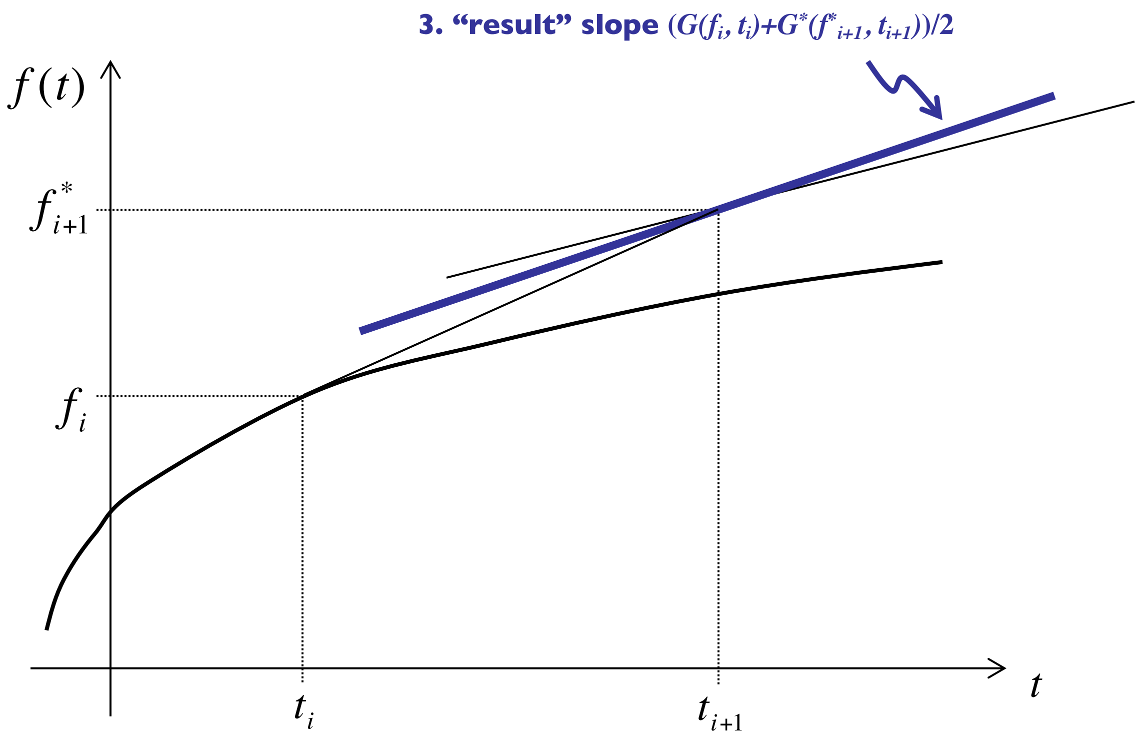
  - modified Euler scheme



Review of Numerical Methods

- 1st order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$

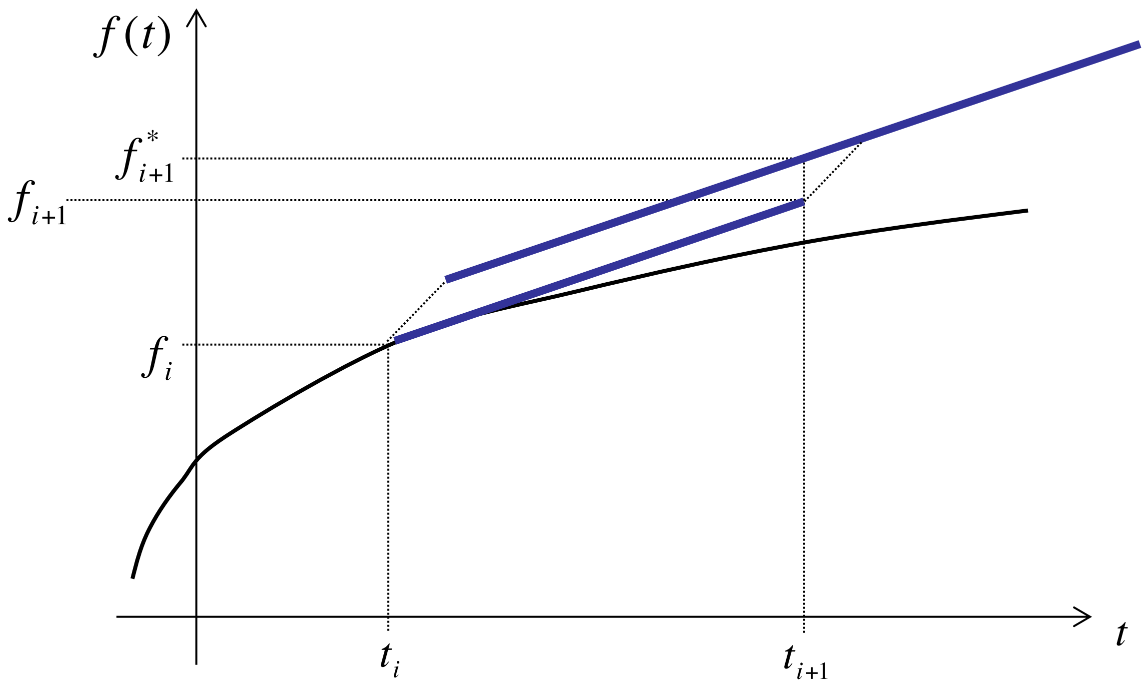
  - modified Euler scheme



Review of Numerical Methods

- 1st order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$

  • modified Euler scheme

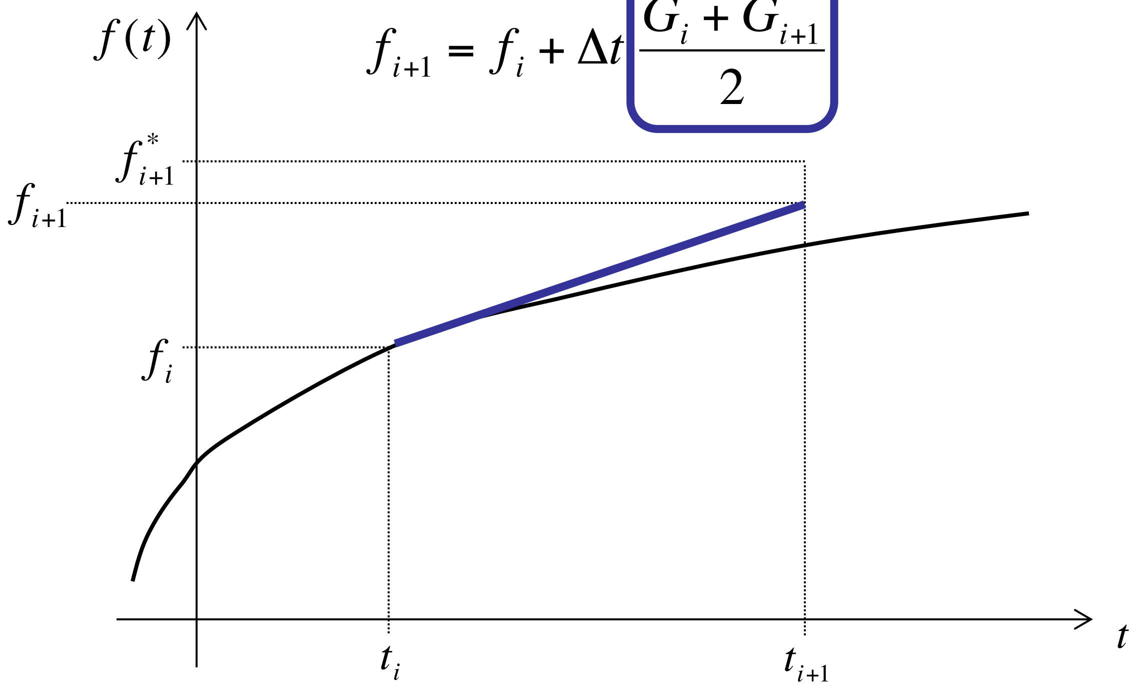


Review of Numerical Methods

- 1[st] order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$

    - modified Euler scheme

$$f_{i+1} = f_i + \Delta t \boxed{\frac{G_i + G^*_{i+1}}{2}}$$



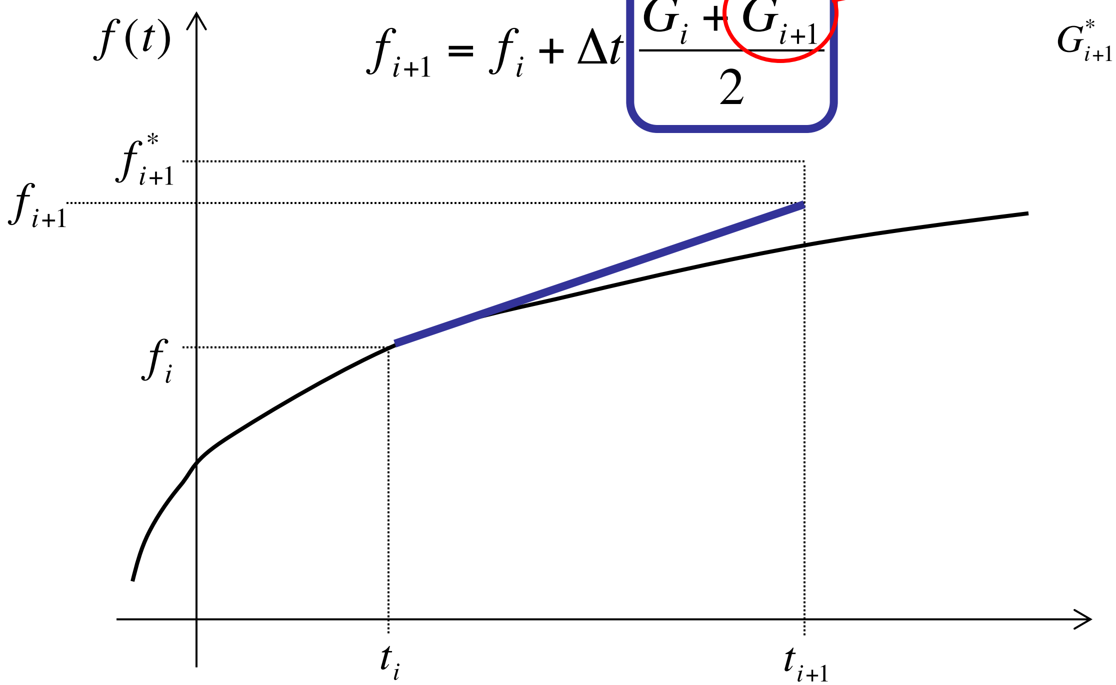
Review of Numerical Methods

- 1st order explicit ordinary differential equation $\frac{df}{dt} = G(f, t)$
  - modified Euler scheme

$$f_{i+1} = f_i + \Delta t \boxed{\frac{G_i + G^*_{i+1}}{2}}$$

additional calculations required!

$$G^*_{i+1} = G(t_i + \Delta t, f_i + \Delta t\, G_i)$$



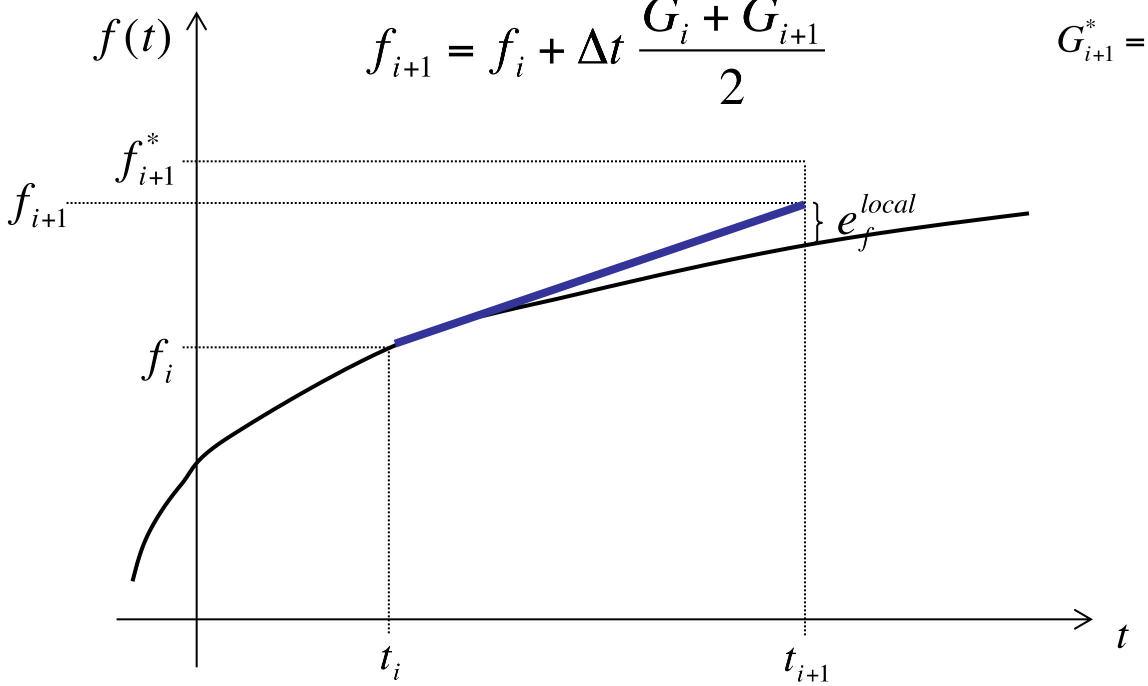
Review of Numerical Methods

- 1^{st} order explicit ordinary differential equation $\dfrac{df}{dt} = G(f,t)$

  - modified Euler scheme

$$f_{i+1} = f_i + \Delta t \, \frac{G_i + G^*_{i+1}}{2}$$

$$G^*_{i+1} = G(t_i + \Delta t, f_i + \Delta t \, G_i)$$



Review of Numerical Methods

- 1st order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$

  - modified Euler scheme

$$f_{i+1} = f_i + \Delta t \, \frac{G_i + G^*_{i+1}}{2}$$

$$G^*_{i+1} = G(t_i + \Delta t, f_i + \Delta t \, G_i)$$

  - local error estimate

Review of Numerical Methods

- 1st order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$

  - modified Euler scheme

$$f_{i+1} = f_i + \Delta t \, \frac{G_i + G^*_{i+1}}{2}$$

$$G^*_{i+1} = G(t_i + \Delta t, f_i + \Delta t \, G_i)$$

  - local error estimate

...eliminate G^*_{i+1}

- 1[st] order explicit ordinary differential equation $\frac{df}{dt} = G(f, t)$

  - modified Euler scheme

$$f_{i+1} = f_i + \Delta t \, \frac{G_i + G^*_{i+1}}{2} \qquad G^*_{i+1} = G(t_i + \Delta t, f_i + \Delta t \, G_i)$$

  - local error estimate

$$\frac{d^2 f}{dt^2} = \dot{G}_i = \frac{G^*_{i+1} - G_i}{\Delta t}$$

Review of Numerical Methods

- 1st order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$

  - modified Euler scheme

$$f_{i+1} = f_i + \Delta t \, \frac{G_i + G^*_{i+1}}{2}$$

$$G^*_{i+1} = G(t_i + \Delta t, f_i + \Delta t \, G_i)$$

  - local error estimate

$$\dot{G}_i = \frac{G^*_{i+1} - G_i}{\Delta t} \quad \Rightarrow \quad G^*_{i+1} = \Delta t \dot{G}_i + G_i$$

- 1^{st} order explicit ordinary differential equation $\frac{df}{dt} = G(f, t)$

  - modified Euler scheme

$$f_{i+1} = f_i + \Delta t \, \frac{G_i + G^*_{i+1}}{2}$$

$$G^*_{i+1} = G(t_i + \Delta t, f_i + \Delta t \, G_i)$$

  - local error estimate

$$\dot{G}_i = \frac{G^*_{i+1} - G_i}{\Delta t} \quad \Rightarrow \quad G^*_{i+1} = \Delta t \dot{G}_i + G_i$$

$$f_{i+1} = f_i + \Delta t \, G_i + \frac{1}{2}(\Delta t)^2 \dot{G}_i + \ldots$$

Review of Numerical Methods

- 1st order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$

  - modified Euler scheme

$$f_{i+1} = f_i + \Delta t \, \frac{G_i + G^*_{i+1}}{2}$$

$$G^*_{i+1} = G(t_i + \Delta t, \, f_i + \Delta t \, G_i)$$

  - local error estimate

$$\dot{G}_i = \frac{G^*_{i+1} - G_i}{\Delta t} \quad \Rightarrow \quad G^*_{i+1} = \Delta t \dot{G}_i + G_i$$

$$f_{i+1} = f_i + \Delta t \, G_i + \frac{1}{2}(\Delta t)^2 \dot{G}_i + \ldots$$

additional term $\dot{G} = \frac{d^2 f}{dt^2}$

Review of Numerical Methods

- 1st order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$

  - modified Euler scheme

$$f_{i+1} = f_i + \Delta t\,\frac{G_i + G^*_{i+1}}{2}$$

$$G^*_{i+1} = G(t_i + \Delta t,\, f_i + \Delta t\, G_i)$$

  - local error estimate

$$\dot{G}_i = \frac{G^*_{i+1} - G_i}{\Delta t} \quad \Rightarrow \quad G^*_{i+1} = \Delta t \dot{G}_i + G_i$$

$$f_{i+1} = f_i + \Delta t\, G_i + \frac{1}{2}(\Delta t)^2 \dot{G}_i + \ldots$$

$$f(t_i + \Delta t) = f(t_i) + \left.\frac{df}{dt}\right|_{t_i} dt + \frac{1}{2}\left.\frac{d^2 f}{dt^2}\right|_{t_i} \left(dt\right)^2$$

Taylor expansion of $f(t)$ about t_i up to 2nd order...

Review of Numerical Methods

- 1^{st} order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$

  - modified Euler scheme

$$f_{i+1} = f_i + \Delta t \, \frac{G_i + G^*_{i+1}}{2} \qquad G^*_{i+1} = G(t_i + \Delta t, f_i + \Delta t \, G_i)$$

  - local error estimate

$$f_{i+1} = f_i + \Delta t \; G_i + \frac{1}{2}(\Delta t)^2 \dot{G}_i + \ldots$$

comparison to F_{i+1}
=> $e_f^{local} \propto (\Delta t)^3$

Review of Numerical Methods

- 1st order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$

  - modified Euler scheme

$$f_{i+1} = f_i + \Delta t \, \frac{G_i + G_{i+1}^*}{2}$$

$$G_{i+1}^* = G(t_i + \Delta t, f_i + \Delta t \, G_i)$$

  - global error estimate

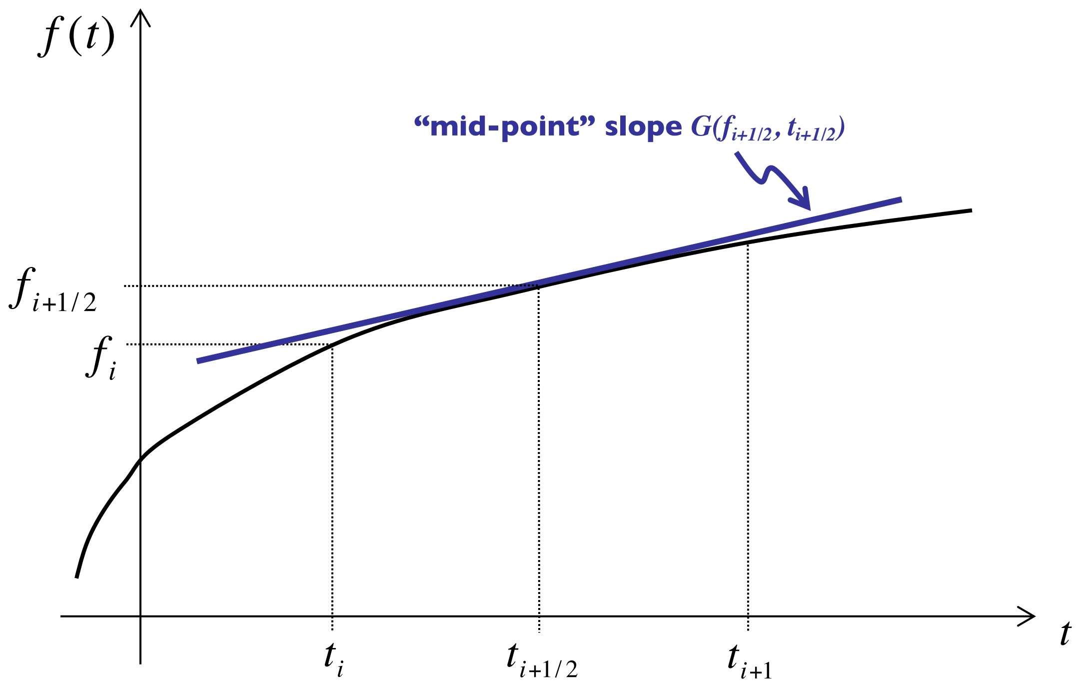
…

=> $e_f^{global} \propto (\Delta t)^2$ "second order accurate"

(at the expanse of more calculations)

Review of Numerical Methods

- 1st order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$
  - 2nd order Runge-Kutta scheme



Review of Numerical Methods

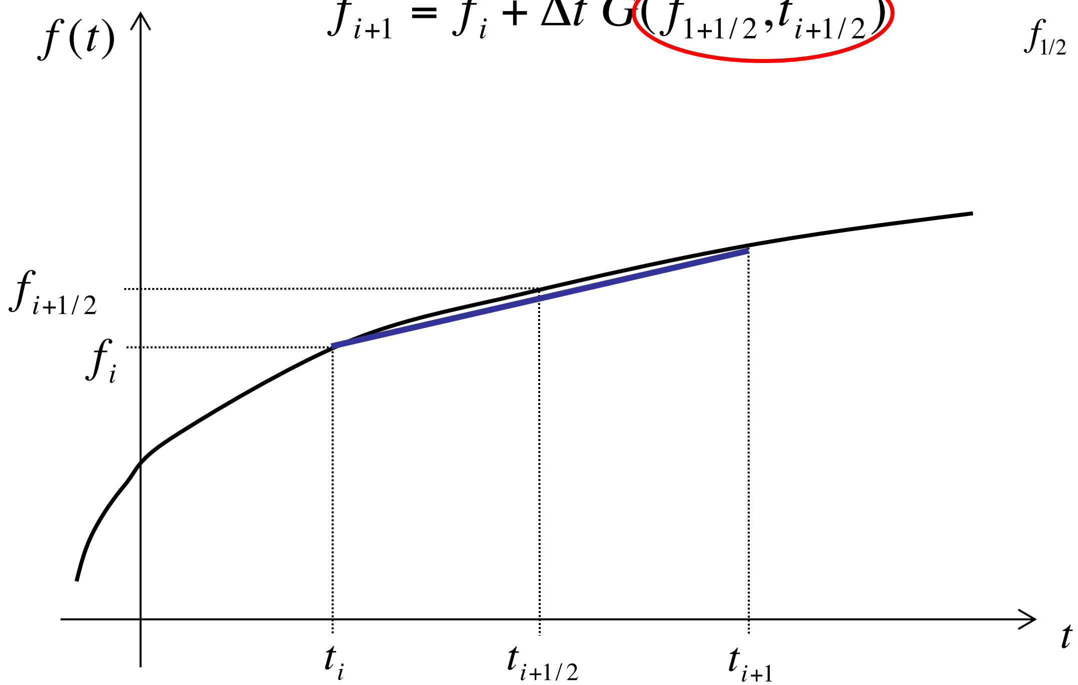
- 1st order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$
  - 2nd order Runge-Kutta scheme

$$f_{i+1} = f_i + \Delta t\, G(f_{1+1/2}, t_{i+1/2})$$

additional calculations required!

$$t_{1/2} = t_i + \Delta t / 2$$

$$f_{1/2} = f_i + \Delta t / 2\, G(f_i, t_i)$$



Review of Numerical Methods

- 1st order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$
  - 2nd order Runge-Kutta scheme

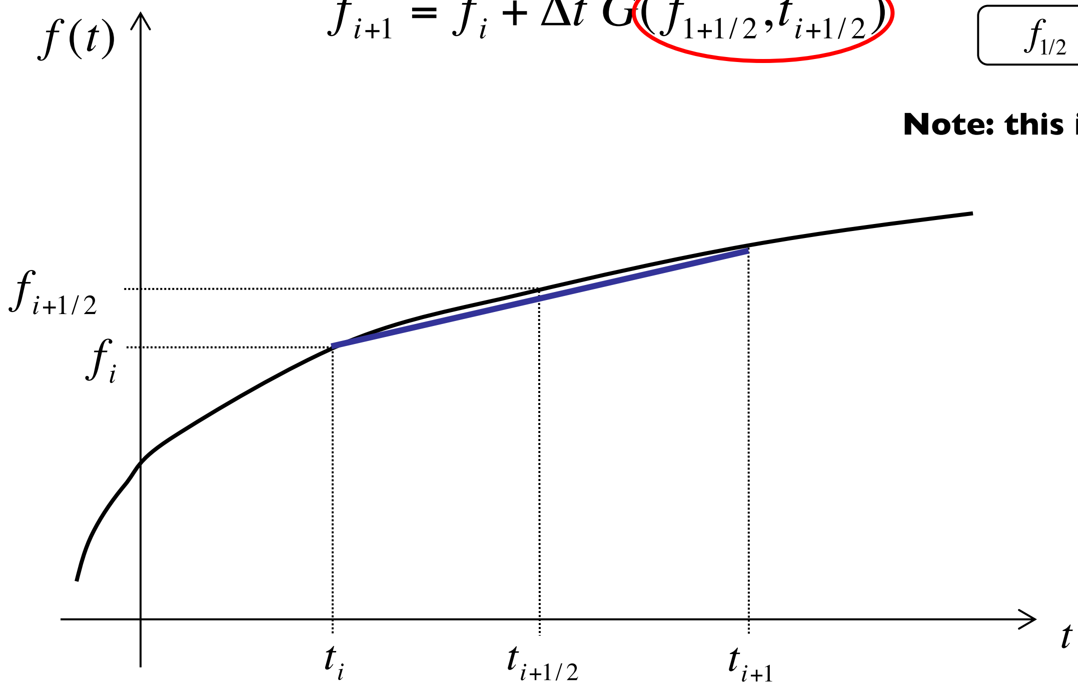
$$f_{i+1} = f_i + \Delta t\, G(f_{1+1/2}, t_{i+1/2})$$

additional calculations required!

$$t_{1/2} = t_i + \Delta t/2$$

$$\boxed{f_{1/2} = f_i + \Delta t/2\, G(f_i, t_i)}$$

Note: this is an Euler step with Δt/2



Review of Numerical Methods

- 1^{st} order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$

  - 2^{nd} order Runge-Kutta scheme

$$f_{i+1} = f_i + \Delta t\, G(f_{1+1/2}, t_{i+1/2})$$

$$t_{1/2} = t_i + \Delta t / 2$$
$$f_{1/2} = f_i + \Delta t / 2\, G(f_i, t_i)$$

  - global error estimate

…

modified Euler & 2^{nd} order Runge-Kutta agree to 1^{st} order (cf. I_3 and I_4 in Numerical Integration)

=> $e_f^{global} \propto (\Delta t)^2$ "second order accurate"

(at the expanse of more calculations)

Review of Numerical Methods

- 1st order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$

  - 4th order Runge-Kutta scheme

$$f_{i+1} = f_i + \Delta t \ \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = G\left(f_i, t_i\right)$$

$$k_2 = G\left(f_i + k_1 \frac{(t_{i+1} - t_i)}{2}, t_i + \frac{(t_{i+1} - t_i)}{2}\right)$$

$$k_3 = G\left(f_i + k_2 \frac{(t_{i+1} - t_i)}{2}, t_i + \frac{(t_{i+1} - t_i)}{2}\right)$$

$$k_4 = G\left(f_i + k_3 (t_{i+1} - t_i), t_{i+1}\right)$$

  - global error estimate

=> $e_f^{global} \propto (\Delta t)^4$ "fourth order accurate"

(at the expanse of *far* more calculations)

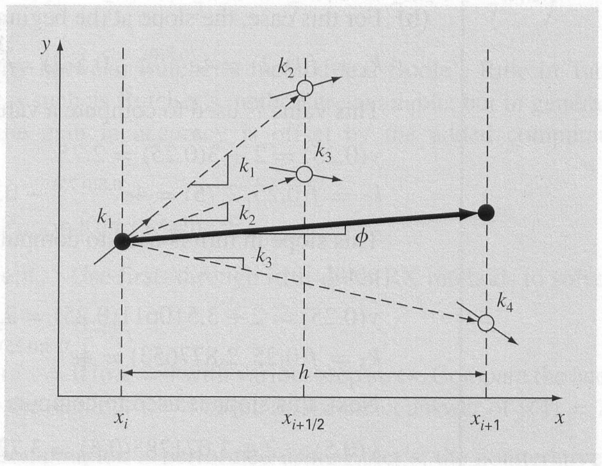
Review of Numerical Methods

- 1^{st} order explicit ordinary differential equation $$\frac{df}{dt} = G(f,t)$$

  - 4^{th} order Runge-Kutta scheme

$$f_{i+1} = f_i + \Delta t \ \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Graphical depiction of the slope estimates comprising the fourth-order RK method.



$$k_1 = G\left(f_i, t_i\right)$$

$$k_2 = G\left(f_i + k_1 \frac{(t_{i+1} - t_i)}{2}, t_i + \frac{(t_{i+1} - t_i)}{2}\right)$$

$$k_3 = G\left(f_i + k_2 \frac{(t_{i+1} - t_i)}{2}, t_i + \frac{(t_{i+1} - t_i)}{2}\right)$$

$$k_4 = G\left(f_i + k_3(t_{i+1} - t_i), t_{i+1}\right)$$

  - global error estimate

=> $e_f^{global} \propto (\Delta t)^4$ "fourth order accurate"

(at the expanse of *far* more calculations)

Review of Numerical Methods

- 1st order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$

  - Euler scheme $$f_{i+1} = f_i + \Delta t\; G(f_i,t_i)$$
  - modified Euler scheme $$f_{i+1} = f_i + \Delta t\; \frac{G_i + G^*_{i+1}}{2}$$
  - 2nd order Runge-Kutta scheme $$f_{i+1} = f_i + \Delta t\; G(f_{1+1/2},t_{i+1/2})$$
  - 4th order Runge-Kutta scheme $$f_{i+1} = f_i + \Delta t\; \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Review of Numerical Methods

- 1^{st} order explicit ordinary differential equation $$\frac{df}{dt} = G(f,t)$$

  - Euler scheme $$f_{i+1} = f_i + \Delta t \, \boxed{G(f_i, t_i)}$$
  - modified Euler scheme $$f_{i+1} = f_i + \Delta t \, \frac{G_i + G_{i+1}^*}{2}$$
  - 2^{nd} order Runge-Kutta scheme $$f_{i+1} = f_i + \Delta t \, G(f_{1+1/2}, t_{i+1/2})$$
  - 4^{th} order Runge-Kutta scheme $$f_{i+1} = f_i + \Delta t \, \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

the only difference lies in the slope to be used…

Review of Numerical Methods

- 1st order explicit ordinary differential equation – Taylor expansion view...

- 1st order explicit ordinary differential equation – Taylor expansion view...

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!} \; (x - x_0)^n$$

Review of Numerical Methods

- 1^{st} order explicit ordinary differential equation – Taylor expansion view...

$$f(x) = f(x_0) + \sum_{n=1}^{\infty} \frac{f^{(n)}(x_0)}{n!} \, (x - x_0)^n$$

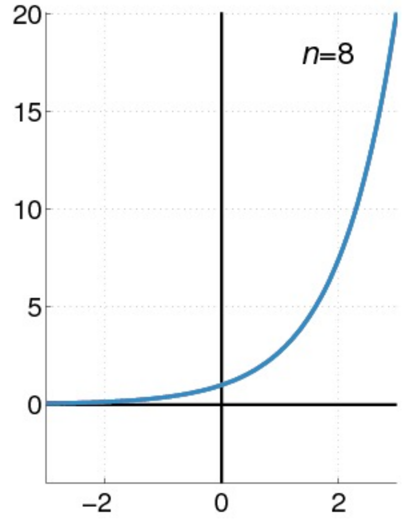
Review of Numerical Methods

- 1[st] order explicit ordinary differential equation – Taylor expansion view...

$$f(x) = f(x_0) + \sum_{n=1}^{\infty} \frac{f^{(n)}(x_0)}{n!} \; (x - x_0)^n$$

+ if we know the function and all its derivatives in x_0 we know the function everywhere

+ the fewer derivatives we know the closer to x_0 this approximation is valid, e.g.

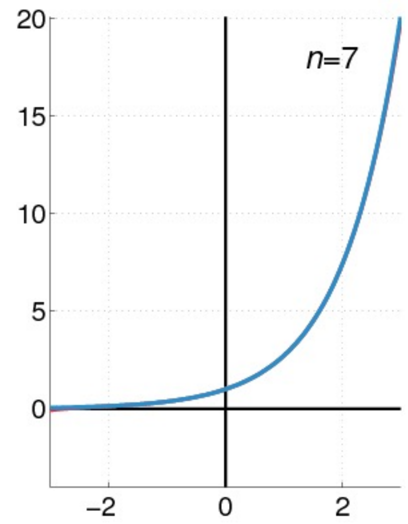


- 1^{st} order explicit ordinary differential equation – Taylor expansion view...

$$f(x) = f(x_0) + \sum_{n=1}^{\infty} \frac{f^{(n)}(x_0)}{n!} (x - x_0)^n$$

+ if we know the function and all its derivatives in x_0 we know the function everywhere

+ the fewer derivatives we know the closer to x_0 this approximation is valid, e.g.



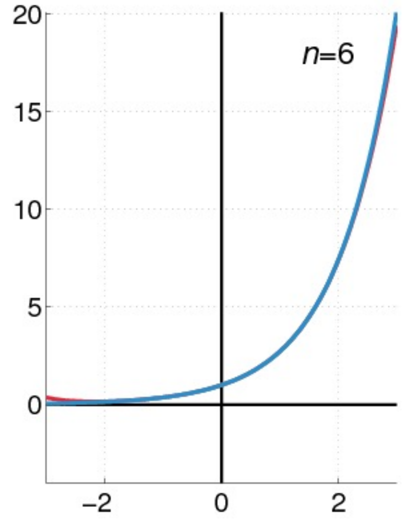
Review of Numerical Methods

- 1st order explicit ordinary differential equation – Taylor expansion view...

$$f(x) = f(x_0) + \sum_{n=1}^{\infty} \frac{f^{(n)}(x_0)}{n!} \, (x - x_0)^n$$

+ if we know the function and all its derivatives in x_0 we know the function everywhere

+ the fewer derivatives we know the closer to x_0 this approximation is valid, e.g.

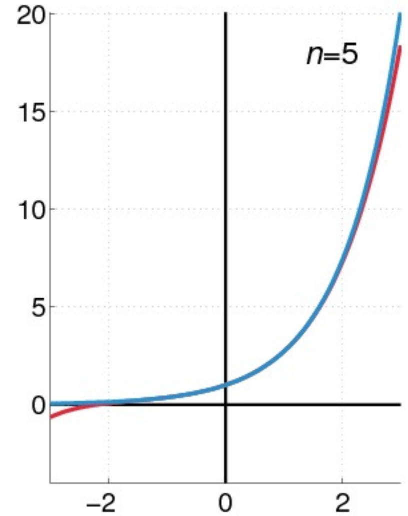


- 1st order explicit ordinary differential equation – Taylor expansion view...

$$f(x) = f(x_0) + \sum_{n=1}^{\infty} \frac{f^{(n)}(x_0)}{n!} \; (x - x_0)^n$$

+ if we know the function and all its derivatives in x_0 we know the function everywhere

+ the fewer derivatives we know the closer to x_0 this approximation is valid, e.g.

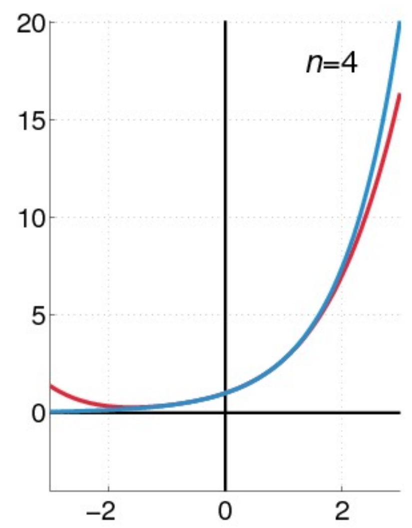


- 1[st] order explicit ordinary differential equation – Taylor expansion view...

$$f(x) = f(x_0) + \sum_{n=1}^{\infty} \frac{f^{(n)}(x_0)}{n!} \, (x - x_0)^n$$

+ if we know the function and all its derivatives in x_0 we know the function everywhere

+ the fewer derivatives we know the closer to x_0 this approximation is valid, e.g.

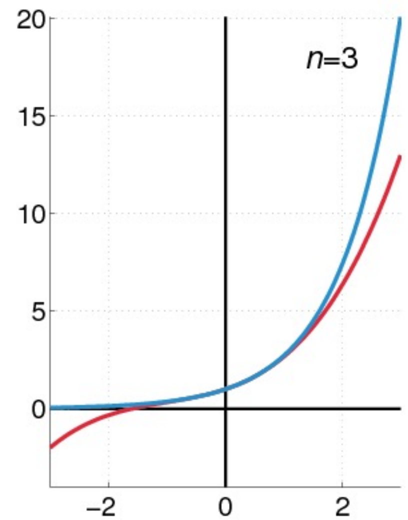


- 1st order explicit ordinary differential equation – Taylor expansion view...

$$f(x) = f(x_0) + \sum_{n=1}^{\infty} \frac{f^{(n)}(x_0)}{n!} (x - x_0)^n$$

+ if we know the function and all its derivatives in x_0 we know the function everywhere

+ the fewer derivatives we know the closer to x_0 this approximation is valid, e.g.



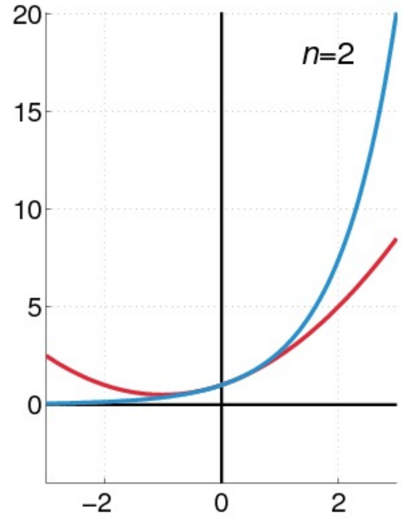
Review of Numerical Methods

- 1^{st} order explicit ordinary differential equation – Taylor expansion view...

$$f(x) = f(x_0) + \sum_{n=1}^{\infty} \frac{f^{(n)}(x_0)}{n!} \; (x - x_0)^n$$

+ if we know the function and all its derivatives in x_0 we know the function everywhere

+ the fewer derivatives we know the closer to x_0 this approximation is valid, e.g.

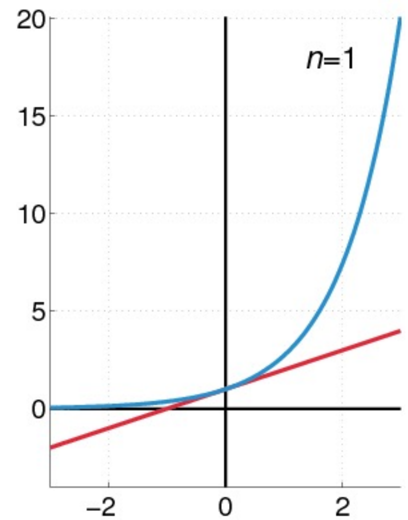


- 1^{st} order explicit ordinary differential equation – Taylor expansion view...

$$f(x) = f(x_0) + \sum_{n=1}^{\infty} \frac{f^{(n)}(x_0)}{n!} (x - x_0)^n$$

+ if we know the function and all its derivatives in x_0 we know the function everywhere

+ the fewer derivatives we know the closer to x_0 this approximation is valid, e.g.

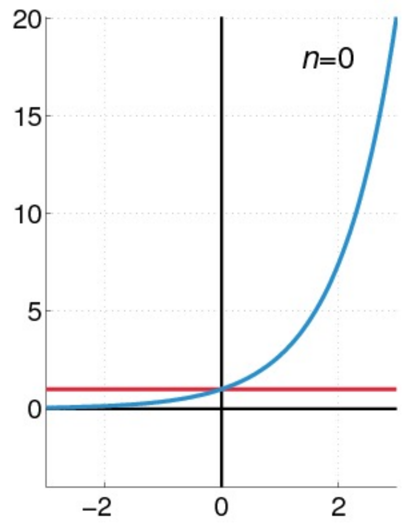


- 1st order explicit ordinary differential equation – Taylor expansion view...

$$f(x) = f(x_0) + \sum_{n=1}^{\infty} \frac{f^{(n)}(x_0)}{n!} \; (x - x_0)^n$$

+ if we know the function and all its derivatives in x_0 we know the function everywhere

+ the fewer derivatives we know the closer to x_0 this approximation is valid, e.g.



Review of Numerical Methods

- 1^{st} order explicit ordinary differential equation – Taylor expansion view...

$$f(x) = f(x_0) + \sum_{n=1}^{\infty} \frac{f^{(n)}(x_0)}{n!} \; (x - x_0)^n$$

+ if we know the function and all its derivatives in x_0 we know the function everywhere

+ the fewer derivatives we know the closer to x_0 this approximation is valid

+ **the Taylor expansion can be used to approximate $f(x)$ in the vicinity of x_0**

- 1^{st} order explicit ordinary differential equation – Taylor expansion view...

$$f(x) = f(x_0) + \sum_{n=1}^{\infty} \frac{f^{(n)}(x_0)}{n!} \, (x - x_0)^n$$

+ if we know the function and all its derivatives in x_0 we know the function everywhere

+ the fewer derivatives we know the closer to x_0 this approximation is valid

+ **the Taylor expansion can be used to approximate *$f(x)$* in the vicinity of x_0:**

$$f(x_0 + \Delta x) = f(x_0) + \sum_{n=1}^{\infty} \frac{f^{(n)}(x_0)}{n!} \, \Delta x^n \qquad ; \ \Delta x = x - x_0$$

- 1[st] order explicit ordinary differential equation – Taylor expansion view...

$$f(x) = f(x_0) + \sum_{n=1}^{\infty} \frac{f^{(n)}(x_0)}{n!} \; (x - x_0)^n$$

this is the key to numerically integrating (differential) equations
where the function and its derivative(s) in a certain point are known...

+ **the Taylor expansion can be used to approximate** $f(x)$ **in the vicinity of** x_0**:**

$$f(x_0 + \Delta x) = f(x_0) + \sum_{n=1}^{\infty} \frac{f^{(n)}(x_0)}{n!} \; \Delta x^n \qquad ; \; \Delta x = x - x_0$$

- 1^{st} order explicit ordinary differential equation $\frac{df}{dt} = G(f,t)$

- Euler scheme $$f_{i+1} = f_i + \Delta t\, G(f_i, t_i)$$

- modified Euler scheme $$f_{i+1} = f_i + \Delta t\, \frac{G_i + G^*_{i+1}}{2}$$

- 2^{nd} order Runge-Kutta scheme $$f_{i+1} = f_i + \Delta t\, G(f_{1+1/2}, t_{i+1/2})$$

- 4^{th} order Runge-Kutta scheme $$f_{i+1} = f_i + \Delta t\, \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

- Taylor expansion $$f(t_0 + \Delta t) = f(t_0) + \sum_{n=1}^{\infty} \frac{f^{(n)}(t_0)}{n!} \Delta t^n$$

- ***1st order !?*** ordinary differential equation $\frac{df}{dt} = G(f,t)$

• Euler scheme $$f_{i+1} = f_i + \Delta t\, G(f_i,t_i)$$

• modified Euler scheme $$f_{i+1} = f_i + \Delta t\, \frac{G_i + G^*_{i+1}}{2}$$

• 2nd order Runge-Kutta scheme $$f_{i+1} = f_i + \Delta t\, G(f_{1+1/2},t_{i+1/2})$$

• 4th order Runge-Kutta scheme $$f_{i+1} = f_i + \Delta t\, \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

• Taylor expansion $$f(t_0 + \Delta t) = f(t_0) + \sum_{n=1}^{\infty} \frac{f^{(n)}(t_0)}{n!}\Delta t^n$$

Review of Numerical Methods

- 2^{nd} order explicit ordinary differential equation

$$\frac{d^2 f}{dt^2} = G(\dot{f}, f, t)$$

- 2nd order explicit ordinary differential equation

$$\frac{df}{dt} = h(f,h,t)$$
$$\frac{dh}{dt} = g(f,h,t)$$

single 2nd order equation $\longrightarrow$

$\longleftarrow$ *system of two 1st order equations*

$$\frac{d^2 f}{dt^2} = G(\dot{f},f,t)$$

Review of Numerical Methods

- 2^{nd} order explicit ordinary differential equation

$$\underbrace{\begin{aligned} \frac{df}{dt} &= h(f,h,t) \\ \frac{dh}{dt} &= g(f,h,t) \end{aligned}}_{\textbf{solve using schemes for 1}^{\textbf{st}}\textbf{ order equations}} \quad \underset{\textit{system of two } 1^{st} \textit{ order equations}}{\overset{\textit{single } 2^{nd} \textit{ order equation}}{\rightleftarrows}} \quad \frac{d^2 f}{dt^2} = G(\dot{f},f,t)$$

Review of Numerical Methods

- 2nd order explicit ordinary differential equation

$$\frac{df}{dt} = h(f,h,t)$$
$$\frac{dh}{dt} = g(f,h,t)$$

single 2nd order equation $\longrightarrow$

$\longleftarrow$ *system of two 1st order equations*

$$\frac{d^2 f}{dt^2} = G(\dot{f}, f, t)$$

solve using schemes for 1st order equations:

- Euler method

$$f_{i+1} = f_i + h(f_i, h_i, t_i)\,\Delta t$$
$$h_{i+1} = h_i + g(f_i, h_i, t_i)\,\Delta t$$

- 2nd order Runge-Kutta

$$t_{mid} = t_i + \Delta t / 2$$
$$f_{mid} = f_i + h(f_i, h_i, t_i)\,\Delta t / 2$$
$$h_{mid} = h_i + g(f_i, h_i, t_i)\,\Delta t / 2$$

$$f_{i+1} = f_i + h(f_{mid}, h_{mid}, t_{mid})\,\Delta t$$
$$h_{i+1} = h_i + g(f_{mid}, h_{mid}, t_{mid})\,\Delta t$$

- 2^{nd} order explicit ordinary differential equation

$$\frac{df}{dt} = h(f,h,t)$$
$$\frac{dh}{dt} = g(f,h,t)$$

single 2^{nd} order equation $\longrightarrow$

$\longleftarrow$ *system of two 1^{st} order equations*

$$\frac{d^2 f}{dt^2} = G(\dot{f}, f, t)$$

solve using schemes for 1^{st} order equations:

- Euler method

$$f_{i+1} = f_i + h(f_i, h_i, t_i)\ \Delta t$$

both variables f and h have to be advanced to the mid-point before making the actual integration step

- 2^{nd} order Runge-Kutta

$$t_{mid} = t_i + \Delta t/2$$
$$\begin{cases} f_{mid} = f_i + h(f_i, h_i, t_i)\ \Delta t/2 \\ h_{mid} = h_i + g(f_i, h_i, t_i)\ \Delta t/2 \end{cases}$$

$$f_{i+1} = f_i + h(f_{mid}, h_{mid}, t_{mid})\ \Delta t$$
$$h_{i+1} = h_i + g(f_{mid}, h_{mid}, t_{mid})\ \Delta t$$

- 2^{nd} order explicit ordinary differential equation

$$\underbrace{\begin{aligned} \frac{df}{dt} &= h(f,h,t) \\ \frac{dh}{dt} &= g(f,h,t) \end{aligned}}$$

single 2^{nd} order equation $\longrightarrow$

$\longleftarrow$ *system of two 1^{st} order equations*

$$\frac{d^2 f}{dt^2} = G(\dot{f}, f, t)$$

solve using schemes for 1^{st} order equations:

- 4^{th} order Runge-Kutta (coefficients)

$$k_1 = h(f_i, h_i, t_i)$$
$$m_1 = g(f_i, h_i, t_i)$$

$$k_2 = h\left(f_i + k_1\frac{(t_{i+1} - t_i)}{2}, h_i + m_1\frac{(t_{i+1} - t_i)}{2}, t_i\right)$$
$$m_2 = g\left(f_i + k_1\frac{(t_{i+1} - t_i)}{2}, h_i + m_1\frac{(t_{i+1} - t_i)}{2}, t_i\right)$$

...

- 2^{nd} order Runge-Kutta

$$t_{mid} = t_i + \Delta t/2$$
$$f_{mid} = f_i + h(f_i, h_i, t_i)\ \Delta t/2$$
$$h_{mid} = h_i + g(f_i, h_i, t_i)\ \Delta t/2$$

$$f_{i+1} = f_i + h(f_{mid}, h_{mid}, t_{mid})\ \Delta t$$
$$h_{i+1} = h_i + g(f_{mid}, h_{mid}, t_{mid})\ \Delta t$$

- 2^{nd} order explicit ordinary differential equation

$$\frac{df}{dt} = h(f,h,t)$$
$$\frac{dh}{dt} = g(f,h,t)$$

single 2^{nd} order equation →

← *system of two 1^{st} order equations*

$$\frac{d^2 f}{dt^2} = G(\dot{f},f,t)$$

solve using schemes for 1^{st} order equations:

- 4^{th} order Runge-Kutta (coefficients)

$$k_1 = h(f_i, h_i, t_i)$$
$$m_1 = g(f_i, h_i, t_i)$$

$$k_2 = h\left(f_i + k_1\frac{(t_{i+1}-t_i)}{2}, h_i + m_1\frac{(t_{i+1}-t_i)}{2}, t_i\right)$$
$$m_2 = g\left(f_i + k_1\frac{(t_{i+1}-t_i)}{2}, h_i + m_1\frac{(t_{i+1}-t_i)}{2}, t_i\right)$$

...

- 2^{nd} order Runge-Kutta

$$t_{mid} = t_i + \Delta t/2$$
$$f_{mid} = f_i + h(f_i,h_i,t_i)\Delta t/2$$
$$h_{mid} = h_i + g(f_i,h_i,t_i)\Delta t/2$$

$$f_{i+1} = f_i + h(f_{mid},h_{mid},t_{mid})\Delta t$$
$$h_{i+1} = h_i + g(f_{mid},h_{mid},t_{mid})\Delta t$$

analytical functions of *f,h,t*

- 2^{nd} order explicit ordinary differential equation

$$\underbrace{\begin{aligned}\frac{df}{dt} &= h(f,h,t)\\ \frac{dh}{dt} &= g(f,h,t)\end{aligned}}$$

single 2^{nd} order equation →

← *system of two 1^{st} order equations*

$$\frac{d^2 f}{dt^2} = G(\dot{f}, f, t)$$

solve using schemes for 1^{st} order equations:

- 4^{th} order Runge-Kutta (coefficients)

higher order schemes require...

...additional calculations

...additional storage

$$\boxed{k_1} = h(f_i, h_i, t_i)$$
$$\boxed{m_1} = g(f_i, h_i, t_i)$$

$$\boxed{k_2} = h\left(f_i + k_1\frac{(t_{i+1}-t_i)}{2}, h_i + m_1\frac{(t_{i+1}-t_i)}{2}, t_i\right)$$
$$\boxed{m_2} = g\left(f_i + k_1\frac{(t_{i+1}-t_i)}{2}, h_i + m_1\frac{(t_{i+1}-t_i)}{2}, t_i\right)$$

...

- 2^{nd} order Runge-Kutta

$$\boxed{t_{mid}} = t_i + \Delta t/2$$
$$\boxed{f_{mid}} = f_i + h(f_i, h_i, t_i)\ \Delta t/2$$
$$\boxed{h_{mid}} = h_i + g(f_i, h_i, t_i)\ \Delta t/2$$

$$f_{i+1} = f_i + h(f_{mid}, h_{mid}, t_{mid})\ \Delta t$$
$$h_{i+1} = h_i + g(f_{mid}, h_{mid}, t_{mid})\ \Delta t$$

- 2nd order explicit ordinary differential equation

$$\frac{df}{dt} = h(f,h,t) \qquad \frac{dh}{dt} = g(f,h,t)$$

single 2nd order equation →

← *system of two 1st order equations*

$$\frac{d^2 f}{dt^2} = G(\dot{f}, f, t)$$

solve using schemes for 1st order equations:

- 4th order Runge-Kutta (coefficients)

higher order sch

maybe not all!?!?!

...additional storage

$$k_1 = h(f_i, h_i,$$
$$m_1 = g(f_i, h_i$$

$$k_2 = h\left(f_i + k_1 \frac{(t_{i+1} - t_i)}{2}, h_i + m_1 \frac{(t_{i+1} - t_i)}{2}, t_i\right)$$
$$m_2 = g\left(f_i + k_1 \frac{(t_{i+1} - t_i)}{2}, h_i + m_1 \frac{(t_{i+1} - t_i)}{2}, t_i\right)$$

...

- 2nd order Runge-Kutta

$$t_{mid} = t_i + \Delta t/2$$
$$f_{mid} = f_i + h(f_i, h_i, t_i)\,\Delta t/2$$
$$h_{mid} = h_i + g(f_i, h_i, t_i)\,\Delta t/2$$

$$f_{i+1} = f_i + h(f_{mid}, h_{mid}, t_{mid})\,\Delta t$$
$$h_{i+1} = h_i + g(f_{mid}, h_{mid}, t_{mid})\,\Delta t$$

Review of Numerical Methods

- 2nd order explicit ordinary differential equation

  - leap-frog scheme

$$\frac{d^2 f}{dt^2} = G(\dot{f}, f, t)$$

- 2^{nd} order explicit ordinary differential equation

  - leap-frog scheme

$$\frac{d^2 f}{dt^2} = G(\cancel{\dot{f}}, f, \cancel{t})$$

- 2^{nd} order explicit ordinary differential equation
  - leap-frog scheme

$$\frac{d^2 f}{dt^2} = G(f)$$

- 2^{nd} order explicit ordinary differential equation

  - leap-frog scheme

$$\frac{df}{dt} = h\,, \quad \frac{dh}{dt} = G(f)$$

rhs only depends on h rhs only depends on f

- 2^{nd} order explicit ordinary differential equation

  - leap-frog scheme

$$\frac{df}{dt} = h\,,\quad \frac{dh}{dt} = G(f)$$

rhs only depends on h rhs only depends on f

$$f_{i+1} = f_i \qquad + \Delta t\, h_{i+1/2}$$

$$h_{i+3/2} = h_{i+1/2} + \Delta t\, G(f_{i+1})$$

- 2^{nd} order explicit ordinary differential equation

  - leap-frog scheme

$$\frac{df}{dt} = h\,,\quad \frac{dh}{dt} = G(f)$$

$$f_{i+1} = f_i \qquad + \Delta t \boxed{h_{i+1/2}}$$

$$h_{i+3/2} = h_{i+1/2} + \Delta t \boxed{G(f_{i+1})}$$

df/dt* only depends on *h

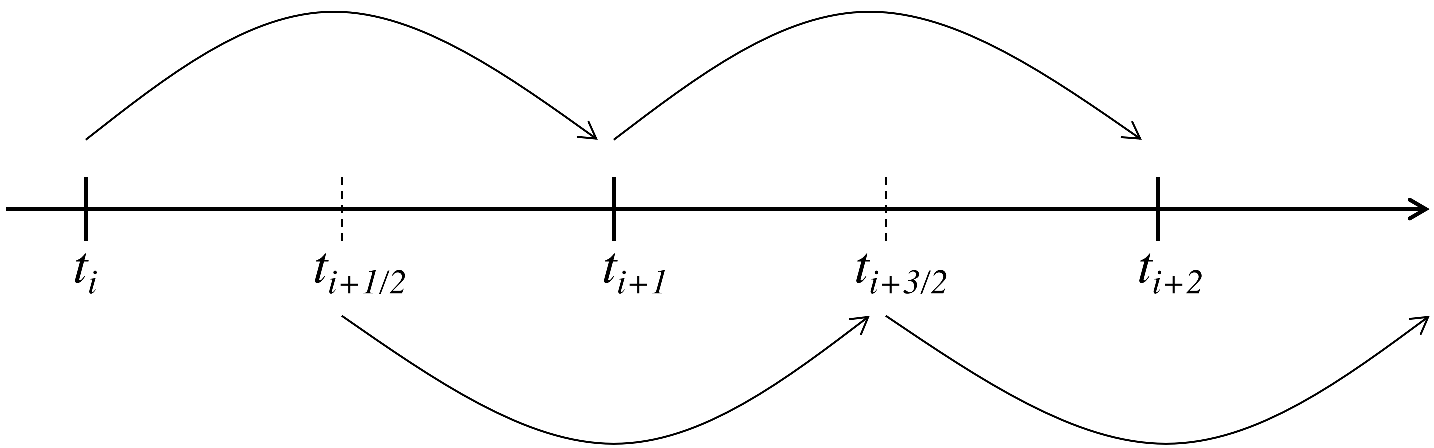
dh/dt* only depends on *f

- 2nd order explicit ordinary differential equation

  - leap-frog scheme

$$\frac{d^2 f}{dt^2} = G(f)$$

$$f_{i+1} = f_i + \Delta t \, h_{i+1/2}$$



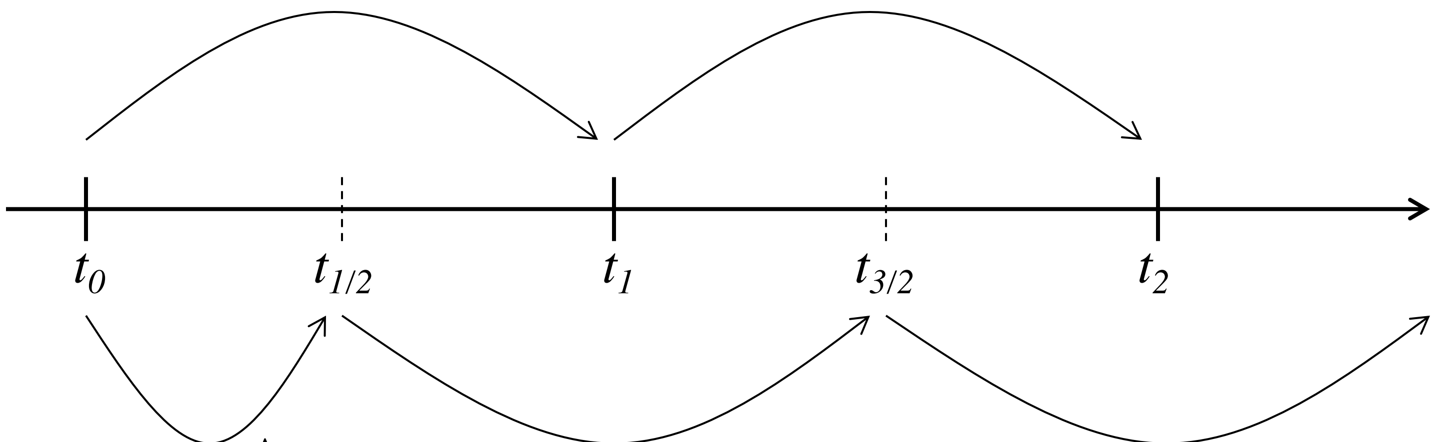
$$h_{i+3/2} = h_{i+1/2} + \Delta t \, G(f_{i+1})$$

- 2nd order explicit ordinary differential equation

  - leap-frog scheme - jumpstart

$$\frac{d^2 f}{dt^2} = G(f)$$

$$f_1 = f_0 + \Delta t \, h_{1/2}$$



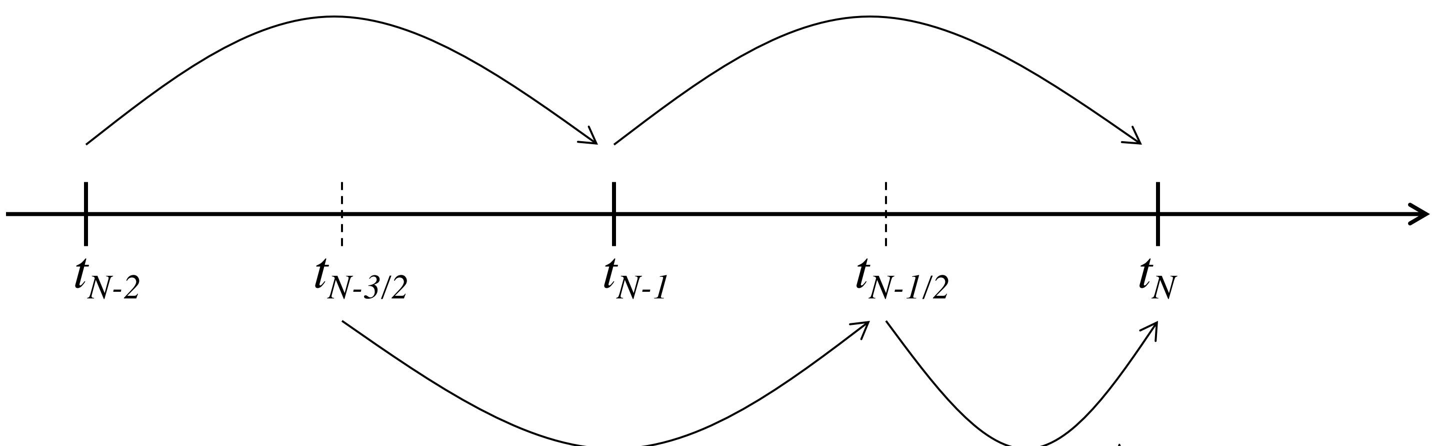
$$h_{1/2} = h_0 + \frac{\Delta t}{2} G(f_0)$$

- 2^{nd} order explicit ordinary differential equation

  - leap-frog scheme - resync

$$\frac{d^2 f}{dt^2} = G(f)$$

$$f_N = f_{N-1} + \Delta t \, h_{N-1/2}$$



$$h_N = h_{N-1/2} + \frac{\Delta t}{2} G(f_N)$$

- 2^{nd} order explicit ordinary differential equation

  - leap-frog scheme

$$\frac{d^2 f}{dt^2} = G(f) \longrightarrow \begin{aligned} f_{i+1} &= f_i + \Delta t\, h_{i+1/2} \\ h_{i+3/2} &= h_{i+1/2} + \Delta t\, G(f_{i+1}) \end{aligned}$$

+ second order accurate scheme

+ no additional calculations or storage

+ symplectic scheme (energy conservation…)

+ very well suited for systems of type $\dfrac{d^2 f}{dt^2} = G(f)$

- 2^{nd} order explicit ordinary differential equation

• leap-frog scheme

$$\frac{d^2 f}{dt^2} = G(f) \longrightarrow \begin{aligned} f_{i+1} &= f_i + \Delta t\, h_{i+1/2} \\ h_{i+3/2} &= h_{i+1/2} + \Delta t\, G(f_{i+1}) \end{aligned}$$

+ second order accurate scheme

+ no additional calculations or storage

+ symplectic scheme (energy conservation…)

+ very well suited for systems of type $\frac{d^2 f}{dt^2} = G(f)$

more accurate than 4^{th} order Runeg-Kutta scheme!?

- 2^{nd} order explicit ordinary differential equation

• leap-frog scheme

$$\frac{d^2 f}{dt^2} = G(f) \longrightarrow \begin{aligned} f_{i+1} &= f_i + \Delta t\, h_{i+1/2} \\ h_{i+3/2} &= h_{i+1/2} + \Delta t\, G(f_{i+1}) \end{aligned}$$

\+ second order accurate scheme

\+ no additional calculations or storage

\+ **symplectic scheme** (energy conservation…)

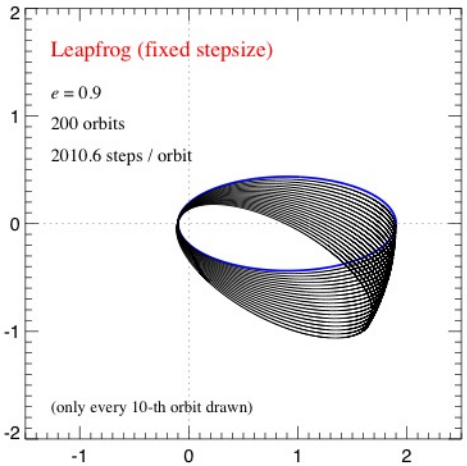
\+ very well suited for systems of type $\dfrac{d^2 f}{dt^2} = G(f)$

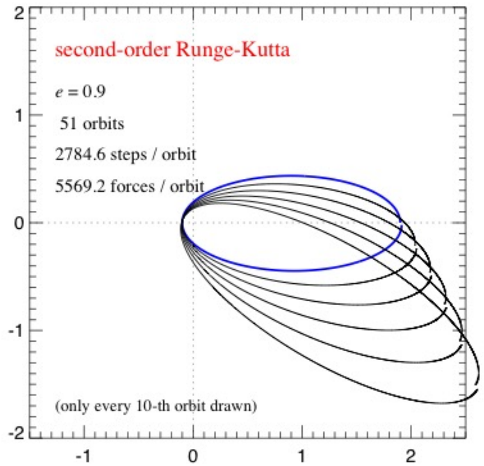
more accurate than 4^{th} order Runeg-Kutta scheme!

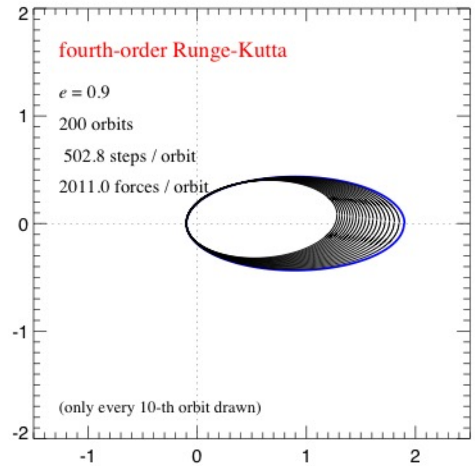
- the basics
- solving differential equations
- **symplectic integration**
- accuracy checks

- symplectic integrators

    - motivation:







symplectic integrator,
2n order accurate

non-symplectic integrators,
2nd order accurate 4th order accurate

taken from Volker Springel's GADGET-2 paper (astro-ph/0505010)

- symplectic integrators

the (numerical) evolution of the system (i.e. x and p) from t_0 to t_n

$$\begin{matrix} \vec{x}_0 \\ \vec{p}_0 \end{matrix} \longrightarrow \begin{matrix} \vec{x}_n \\ \vec{p}_n \end{matrix}$$

is a canonical coordinate transformation

"symplectic" says nothing about the accuracy of the integrator,
but rather preserves the geometric structure of the original Hamiltonian flow!

- symplectic integrators

$$\mathcal{H} = \frac{1}{2m} p^2 + m\Phi(\vec{x})$$

  - equations of motion:

$$\dot{\vec{x}} = \{\vec{x}, \mathcal{H}\}$$
$$\dot{\vec{p}} = \{\vec{p}, \mathcal{H}\}$$

- symplectic integrators

$$\mathcal{H} = \frac{1}{2m} p^2 + m\Phi(\vec{x})$$

- equations of motion:

$$\dot{\vec{x}} = \{\vec{x}, \mathcal{H}\} = H\ \vec{x}$$
$$\dot{\vec{p}} = \{\vec{p}, \mathcal{H}\} = H\ \vec{p}$$

Poisson bracket = linear operator

Review of Numerical Methods

- symplectic integrators

$$\mathcal{H} = \frac{1}{2m} p^2 + m\Phi(\vec{x})$$

- equations of motion:

$$\dot{\vec{x}} = \{\vec{x}, \mathcal{H}\} = H\,\vec{x} \qquad \xrightarrow{\text{evolution of system}} \qquad \vec{x}(t) = e^{tH}\vec{x}_0$$

$$\dot{\vec{p}} = \{\vec{p}, \mathcal{H}\} = H\,\vec{p} \qquad\qquad \vec{p}(t) = e^{tH}\vec{p}_0$$

Poisson bracket = linear operator

- symplectic integrators

$$\mathcal{H} = \frac{1}{2m} p^2 + m\Phi(\vec{x})$$

- equations of motion:

$$\dot{\vec{x}} = \{\vec{x}, \mathcal{H}\} = H\,\vec{x}$$
$$\dot{\vec{p}} = \{\vec{p}, \mathcal{H}\} = H\,\vec{p}$$

$$\xrightarrow{\text{evolution of system}}$$

$$\vec{x}(t) = e^{tH}\vec{x}_0$$
$$\vec{p}(t) = e^{tH}\vec{p}_0$$

$$\mathcal{H} = \mathcal{H}_p + \mathcal{H}_x = T + V$$

- symplectic integrators

$$\mathcal{H} = \frac{1}{2m} p^2 + m\Phi(\vec{x})$$

- equations of motion:

$$\dot{\vec{x}} = \{\vec{x}, \mathcal{H}\} = H\,\vec{x} \qquad \vec{x}(t) = e^{tH}\vec{x}_0$$

$$\dot{\vec{p}} = \{\vec{p}, \mathcal{H}\} = H\,\vec{p} \qquad \vec{p}(t) = e^{tH}\vec{p}_0$$

evolution of system

$$\mathcal{H} = \mathcal{H}_p + \mathcal{H}_x = T + V$$

separation of potential and kinetic flow possible!?

- symplectic integrators

$$\mathcal{H} = \frac{1}{2m} p^2 + m\Phi(\vec{x})$$

- equations of motion:

$$\dot{\vec{x}} = \{\vec{x}, \mathcal{H}\} = H\,\vec{x} \qquad \xrightarrow{\text{evolution of system}} \qquad \vec{x}(t) = e^{t(T+V)}\vec{x}_0$$

$$\dot{\vec{p}} = \{\vec{p}, \mathcal{H}\} = H\,\vec{p} \qquad\qquad \vec{p}(t) = e^{t(T+V)}\vec{p}_0$$

$$\mathcal{H} = \mathcal{H}_p + \mathcal{H}_x = T + V$$

separation of potential and kinetic flow possible!?

- symplectic integrators

$$\mathcal{H} = \frac{1}{2m} p^2 + m\Phi(\vec{x})$$

- equations of motion:

$$\begin{aligned} \dot{\vec{x}} &= \{\vec{x}, \mathcal{H}\} = H\,\vec{x} \\ \dot{\vec{p}} &= \{\vec{p}, \mathcal{H}\} = H\,\vec{p} \end{aligned} \quad \xrightarrow{\text{evolution of system}} \quad \begin{aligned} \vec{x}(t) &= e^{t(T+V)}\vec{x}_0 \\ \vec{p}(t) &= e^{t(T+V)}\vec{p}_0 \end{aligned}$$

$$\mathcal{H} = \mathcal{H}_p + \mathcal{H}_x = T + V$$

try to split $e^{t(T+V)}$ into something like $e^{tT}e^{tV}$?!

- symplectic integrators

$$\mathcal{H} = \frac{1}{2m} p^2 + m\Phi(\vec{x})$$

- equations of motion:

$$\begin{aligned} \dot{\vec{x}} &= \{\vec{x}, \mathcal{H}\} = H\,\vec{x} \\ \dot{\vec{p}} &= \{\vec{p}, \mathcal{H}\} = H\,\vec{p} \end{aligned} \quad \xrightarrow{\text{evolution of system}} \quad \begin{aligned} \vec{x}(t) &= e^{t(T+V)}\vec{x}_0 \\ \vec{p}(t) &= e^{t(T+V)}\vec{p}_0 \end{aligned}$$

$$\mathcal{H} = \mathcal{H}_p + \mathcal{H}_x = T + V$$

- Baker-Campbell-Hausdorff identity:

$$e^A e^B \neq e^{A+B}$$

$$e^A e^B = e^C \quad \text{with } C = A + B + \frac{1}{2}\{A,B\} + \ldots$$

- symplectic integrators

$$\mathcal{H} = \frac{1}{2m}p^2 + m\Phi(\vec{x})$$

- equations of motion:

$$\dot{\vec{x}} = \{\vec{x},\mathcal{H}\} = H\,\vec{x} \qquad \dot{\vec{p}} = \{\vec{p},\mathcal{H}\} = H\,\vec{p} \quad \xrightarrow{\text{evolution of system}} \quad \vec{x}(t) = e^{t(T+V)}\vec{x}_0 \qquad \vec{p}(t) = e^{t(T+V)}\vec{p}_0$$

$$\mathcal{H} = \mathcal{H}_p + \mathcal{H}_x = T + V$$

- Baker-Campbell-Hausdorff identity:

$$e^{t(T+V)} = e^{tT/2}e^{tV}e^{tT/2} + O(t^3)$$

▪ symplectic integrators

$$\mathcal{H} = \frac{1}{2m} p^2 + m\Phi(\vec{x})$$

- equations of motion:

$$\begin{aligned} \dot{\vec{x}} &= \{\vec{x}, \mathcal{H}\} = H\,\vec{x} \\ \dot{\vec{p}} &= \{\vec{p}, \mathcal{H}\} = H\,\vec{p} \end{aligned} \quad \xrightarrow{\text{evolution of system}} \quad \begin{aligned} \vec{x}(t) &= e^{tT/2} e^{tV} e^{tT/2} \vec{x}_0 + O(t^3) \\ \vec{p}(t) &= e^{tT/2} e^{tV} e^{tT/2} \vec{p}_0 + O(t^3) \end{aligned}$$

$$\mathcal{H} = \mathcal{H}_p + \mathcal{H}_x = T + V$$

because the Hamiltonian can be split into two independent parts*,
we are able to separate the evolution of the system's position and momentum…

*i.e. kinetic energy T and potential energy V

- symplectic integrators

$$\mathcal{H} = \frac{1}{2m} p^2 + m\Phi(\vec{x})$$

- equations of motion:

$$\dot{\vec{x}} = \{\vec{x}, \mathcal{H}\} = H\,\vec{x} \qquad \vec{x}(t) = e^{tT/2} e^{tV} e^{tT/2} \vec{x}_0 + O(t^3)$$

$$\dot{\vec{p}} = \{\vec{p}, \mathcal{H}\} = H\,\vec{p} \qquad \vec{p}(t) = e^{tT/2} e^{tV} e^{tT/2} \vec{p}_0 + O(t^3)$$

evolution of system $\longrightarrow$

$$\mathcal{H} = \mathcal{H}_p + \mathcal{H}_x = T + V$$

we rather approximate the true Hamiltonian
than discretizing the equations of motions
=> **symplectic integration scheme**!

- symplectic integrators

$$\mathcal{H} = \frac{1}{2m} p^2 + m\Phi(\vec{x})$$

- second-order accurate scheme:

$$\vec{x}(t) = e^{tT/2} e^{tV} \left(e^{tT/2} \vec{x}_0 \right)$$

$$\vec{p}(t) = e^{tT/2} e^{tV} \left(e^{tT/2} \vec{p}_0 \right)$$

1. evolve the system for $\Delta t/2$ under $\mathcal{H}_p$

- symplectic integrators

$$\mathcal{H} = \frac{1}{2m} p^2 + m\Phi(\vec{x})$$

- second-order accurate scheme:

$$\vec{x}(t) = e^{tT/2}\left(e^{tV}\left(e^{tT/2}\vec{x}_0\right)\right)$$

$$\vec{p}(t) = e^{tT/2}\left(e^{tV}\left(e^{tT/2}\vec{p}_0\right)\right)$$

1. evolve the system for $\Delta t/2$ under $\mathcal{H}_p$
2. evolve the system for Δt under $\mathcal{H}_x$

- symplectic integrators

$$\mathcal{H} = \frac{1}{2m} p^2 + m\Phi(\vec{x})$$

- second-order accurate scheme:

$$\vec{x}(t) = \left(e^{tT/2}\left(e^{tV}\left(e^{tT/2}\vec{x}_0\right)\right)\right)$$

$$\vec{p}(t) = \left(e^{tT/2}\left(e^{tV}\left(e^{tT/2}\vec{p}_0\right)\right)\right)$$

1. evolve the system for $\Delta t/2$ under $\mathcal{H}_p$
2. evolve the system for Δt under $\mathcal{H}_x$
3. evolve the system for $\Delta t/2$ under $\mathcal{H}_p$

- Drift-Kick-Drift time integration

$$\boxed{\begin{array}{ccc} \vec{x}^{\,n} & & \vec{x}^{\,n+1} \\ & \curvearrowright & \\ \vec{p}^{\,n} & & \vec{p}^{\,n+1} \end{array}}$$

superscript indicates t, i.e. $x^{\mathrm{n}} = x(t_{\mathrm{n}})$

- Drift-Kick-Drift time integration

$$\boxed{\begin{array}{ccc} \vec{x}^{\,n} & & \vec{x}^{\,n+1} \\ & \curvearrowright & \\ \vec{p}^{\,n} & & \vec{p}^{\,n+1} \end{array}}$$

$$\dot{\vec{x}} = \{\vec{x}, \mathcal{H}\}$$

$$\dot{\vec{p}} = \{\vec{p}, \mathcal{H}\}$$

superscript indicates t, i.e. $x^{\mathrm{n}} = x(t_{\mathrm{n}})$

Review of Numerical Methods

- Drift-Kick-Drift time integration

$$\begin{array}{ccc} \vec{x}^{\,n} & & \vec{x}^{\,n+1} \\ & \curvearrowright & \\ \vec{p}^{\,n} & & \vec{p}^{\,n+1} \end{array}$$

$$\dot{\vec{x}} = \{\vec{x}, \mathcal{H}\}$$

$$\dot{\vec{p}} = \{\vec{p}, \mathcal{H}\}$$

$$\mathcal{H} = \frac{1}{2m} p^2 + m\Phi(\vec{x})$$

superscript indicates t, i.e. $x^n = x(t_n)$

Review of Numerical Methods

- Drift-Kick-Drift time integration

$$\boxed{\begin{array}{ccc}\vec{x}^{\,n} & & \vec{x}^{\,n+1} \\ & \curvearrowright & \\ \vec{p}^{\,n} & & \vec{p}^{\,n+1}\end{array}}$$

$$\dot{\vec{x}} = \frac{\vec{p}}{m}$$

$$\dot{\vec{p}} = -\nabla\Phi$$

superscript indicates t, i.e. $x^{n} = x(t_{n})$

Review of Numerical Methods

- Drift-Kick-Drift time integration

$$\boxed{\begin{array}{ccc} \vec{x}^{\,n} & & \vec{x}^{\,n+1} \\ & \curvearrowright & \\ \vec{p}^{\,n} & & \vec{p}^{\,n+1} \end{array}}$$

$$\dot{\vec{x}} = \frac{\vec{p}}{m} \qquad => \mathcal{H}_p$$

$$\dot{\vec{p}} = -\nabla\Phi \qquad => \mathcal{H}_x$$

1. evolve the system for $\Delta t/2$ under $\mathcal{H}_p$
2. evolve the system for Δt under $\mathcal{H}_x$
3. evolve the system for $\Delta t/2$ under $\mathcal{H}_p$

superscript indicates t, i.e. $x^{n} = x(t_{n})$

- Drift-Kick-Drift time integration

$$\vec{x}^{\,n} \quad \vec{p}^{\,n} \quad \longrightarrow \quad \vec{x}^{\,n+1} \quad \vec{p}^{\,n+1}$$

$$\dot{\vec{x}} = \frac{\vec{p}}{m}$$

$$\dot{\vec{p}} = -\nabla\Phi$$

symplectic integrators =>

$$\vec{x}^{\,n+1/2} = \vec{x}^{\,n} + \frac{\Delta t}{2}\frac{\vec{p}^{\,n}}{m}$$

$$\vec{p}^{\,n+1} = \vec{p}^{\,n} - \Delta t\,\vec{\nabla}\Phi^{n+1/2}$$

$$\vec{x}^{\,n+1} = \vec{x}^{\,n+1/2} + \frac{\Delta t}{2}\frac{\vec{p}^{\,n+1}}{m}$$

superscript indicates t, i.e. $x^{\mathrm{n}} = x(t_{\mathrm{n}})$

- Drift-Kick-Drift time integration

$$\boxed{\begin{matrix} \vec{x}^{\,n} & & \vec{x}^{\,n+1} \\ & \curvearrowright & \\ \vec{p}^{\,n} & & \vec{p}^{\,n+1} \end{matrix}}$$

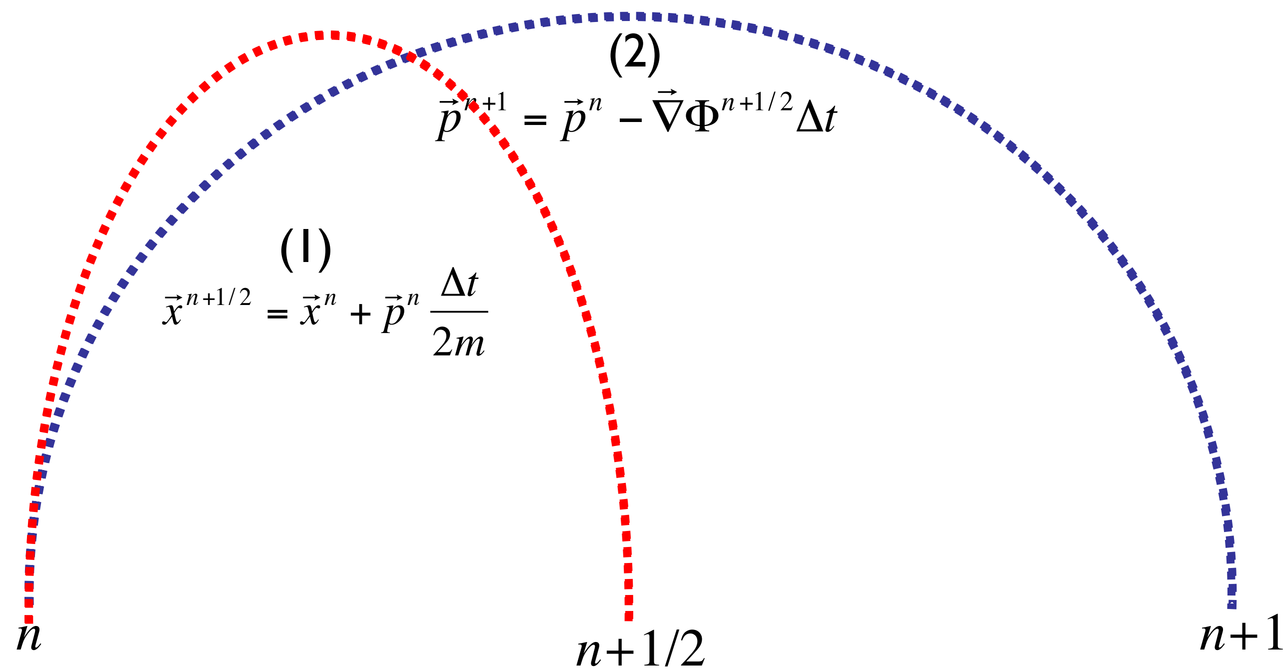
(1)

$$\vec{x}^{\,n+1/2} = \vec{x}^{\,n} + \vec{p}^{\,n}\frac{\Delta t}{2m}$$

n $\quad$ $n+1/2$ $\quad$ $n+1$

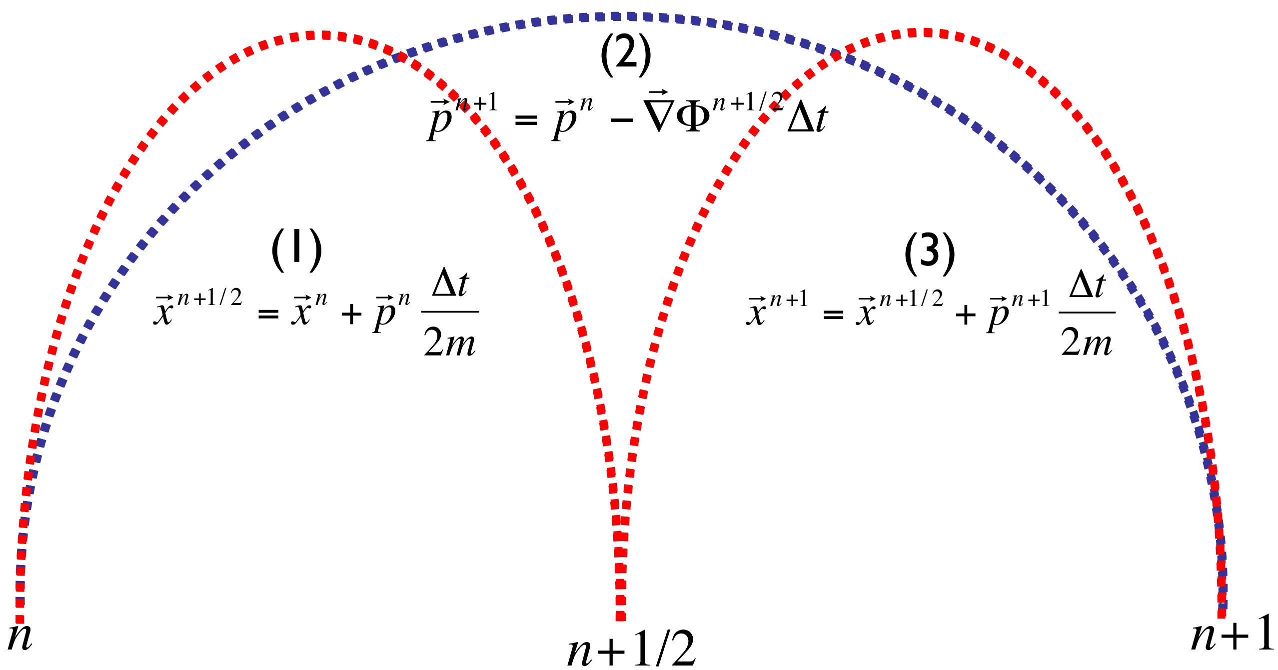
- Drift-Kick-Drift time integration

$$\begin{matrix} \vec{x}^{\,n} & & \vec{x}^{\,n+1} \\ & \curvearrowright & \\ \vec{p}^{\,n} & & \vec{p}^{\,n+1} \end{matrix}$$



- Drift-Kick-Drift time integration

$$\begin{pmatrix} \vec{x}^{\,n} \\ \vec{p}^{\,n} \end{pmatrix} \longrightarrow \begin{pmatrix} \vec{x}^{\,n+1} \\ \vec{p}^{\,n+1} \end{pmatrix}$$



- Drift-Kick-Drift time integration

$$\vec{x}^{\,n} \quad \vec{p}^{\,n} \longrightarrow \vec{x}^{\,n+1} \quad \vec{p}^{\,n+1}$$

(2)

$$\vec{p}^{\,n+1} = \vec{p}^{\,n} - \vec{\nabla}\Phi^{n+1/2}\Delta t$$

(1)

$$\vec{x}^{\,n+1/2} = \vec{x}^{\,n} + \vec{p}^{\,n}\frac{\Delta t}{2m}$$

(3)

$$\vec{x}^{\,n+1} = \vec{x}^{\,n+1/2} + \vec{p}^{\,n+1}\frac{\Delta t}{2m}$$

3N updates per one time step

n $\quad$ $n+1/2$ $\quad$ $n+1$

Review of Numerical Methods

- recovering the leap-frog scheme

n

$$\vec{x}^{\,n+1/2} = \vec{x}^{\,n} + \frac{\Delta t}{2} \frac{\vec{p}^{\,n}}{m}$$

$$\vec{p}^{\,n+1} = \vec{p}^{\,n} - \Delta t \; \vec{\nabla}\Phi^{n+1/2}$$

$$\vec{x}^{\,n+1} = \vec{x}^{\,n+1/2} + \frac{\Delta t}{2} \frac{\vec{p}^{\,n+1}}{m}$$

$n+1$

$$\vec{x}^{\,n+3/2} = \vec{x}^{\,n+1} + \frac{\Delta t}{2} \frac{\vec{p}^{\,n+1}}{m}$$

- recovering the leap-frog scheme

n

$$\vec{x}^{\,n+1/2} = \vec{x}^{\,n} + \frac{\Delta t}{2} \frac{\vec{p}^{\,n}}{m}$$

$$\vec{p}^{\,n+1} = \vec{p}^{\,n} - \Delta t \ \vec{\nabla}\Phi^{n+1/2}$$

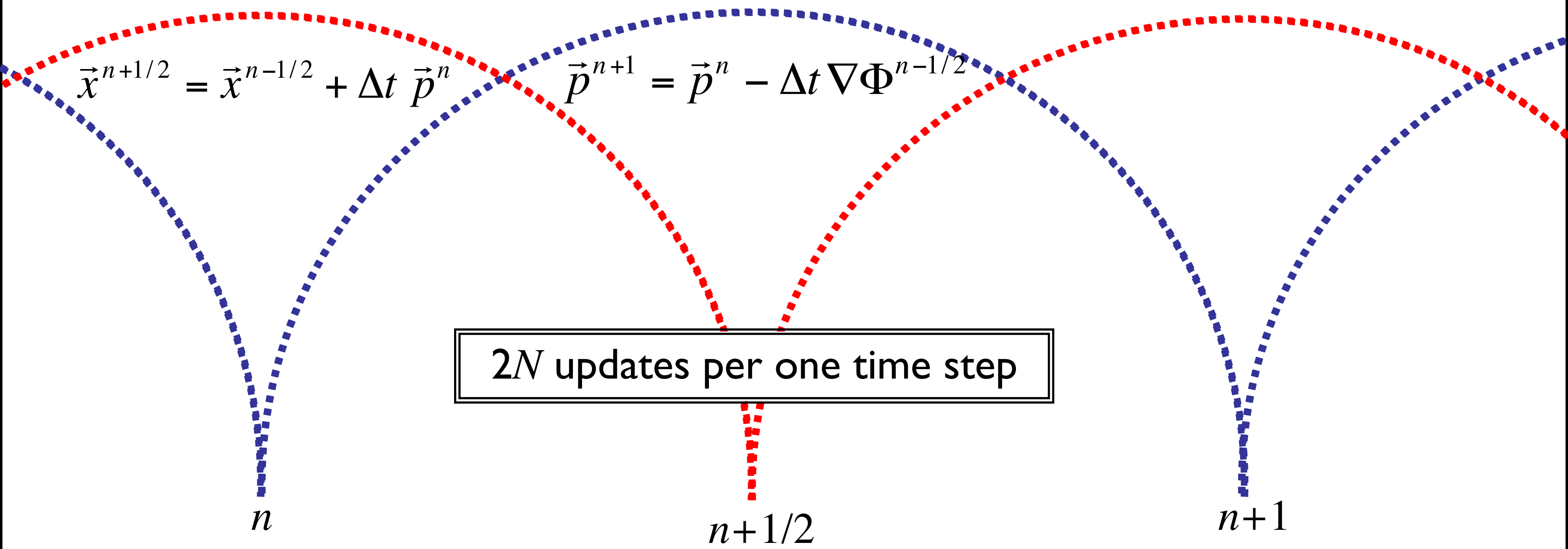
$$\vec{p}^{\,n+1} = \vec{p}^{\,n} - \Delta t \ \vec{\nabla}\Phi^{n+1/2}$$

$$\left.\begin{aligned} \vec{x}^{\,n+1} &= \vec{x}^{\,n+1/2} + \frac{\Delta t}{2} \frac{\vec{p}^{\,n+1}}{m} \\ &\underline{n+1} \\ \vec{x}^{\,n+3/2} &= \vec{x}^{\,n+1} + \frac{\Delta t}{2} \frac{\vec{p}^{\,n+1}}{m} \end{aligned}\right\} \quad \vec{x}^{\,n+3/2} = \vec{x}^{\,n+1/2} + \Delta t \ \frac{\vec{p}^{\,n+1}}{m}$$

- recovering the leap-frog scheme

$$\boxed{\begin{matrix} \vec{x}^{\,n} & & \vec{x}^{\,n+1} \\ & \curvearrowright & \\ \vec{p}^{\,n} & & \vec{p}^{\,n+1} \end{matrix}}$$

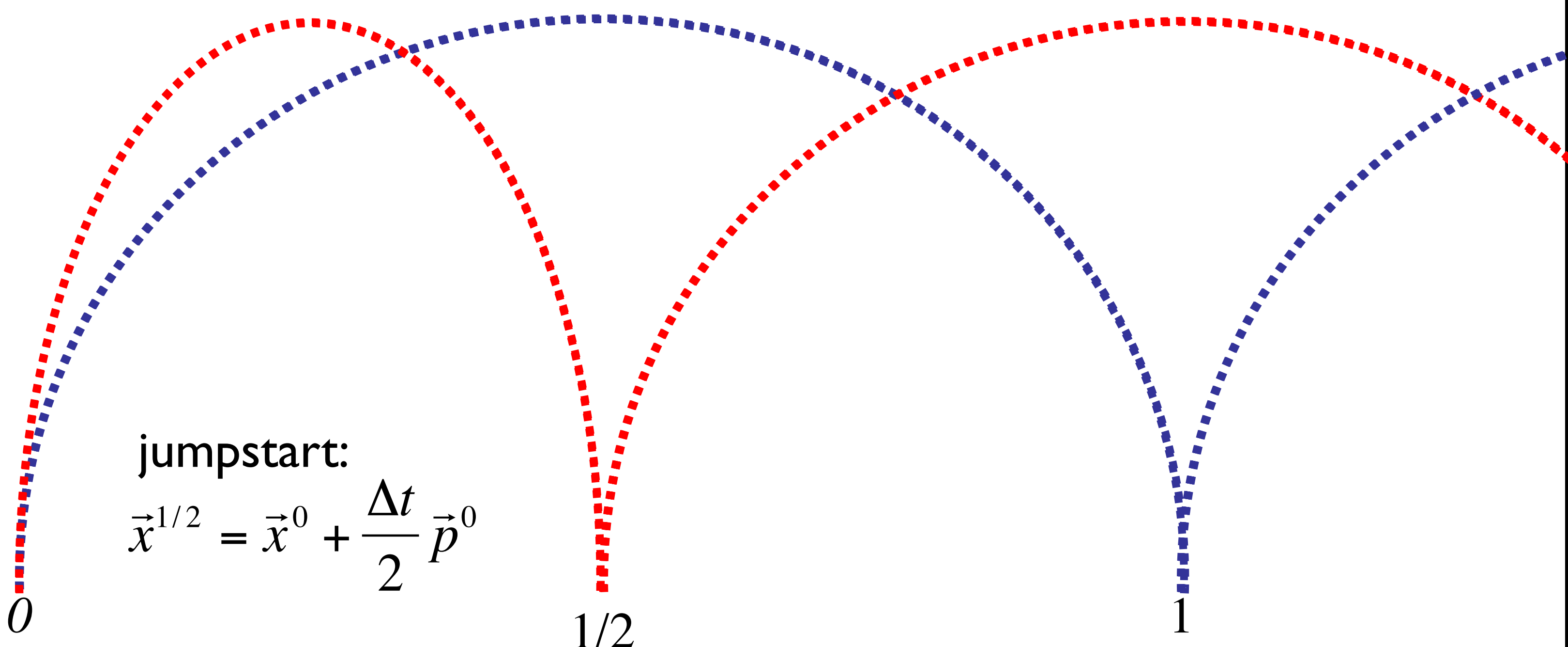
$$\vec{x}^{\,n+1/2} = \vec{x}^{\,n-1/2} + \Delta t\ \vec{p}^{\,n} \qquad \vec{p}^{\,n+1} = \vec{p}^{\,n} - \Delta t\, \nabla\Phi^{n-1/2}$$



$\boxed{2N \text{ updates per one time step}}$

- recovering the leap-frog scheme

$$\boxed{\begin{array}{ccc} \vec{x}^{\,n} & & \vec{x}^{\,n+1} \\ \vec{p}^{\,n} & \curvearrowright & \vec{p}^{\,n+1} \end{array}}$$

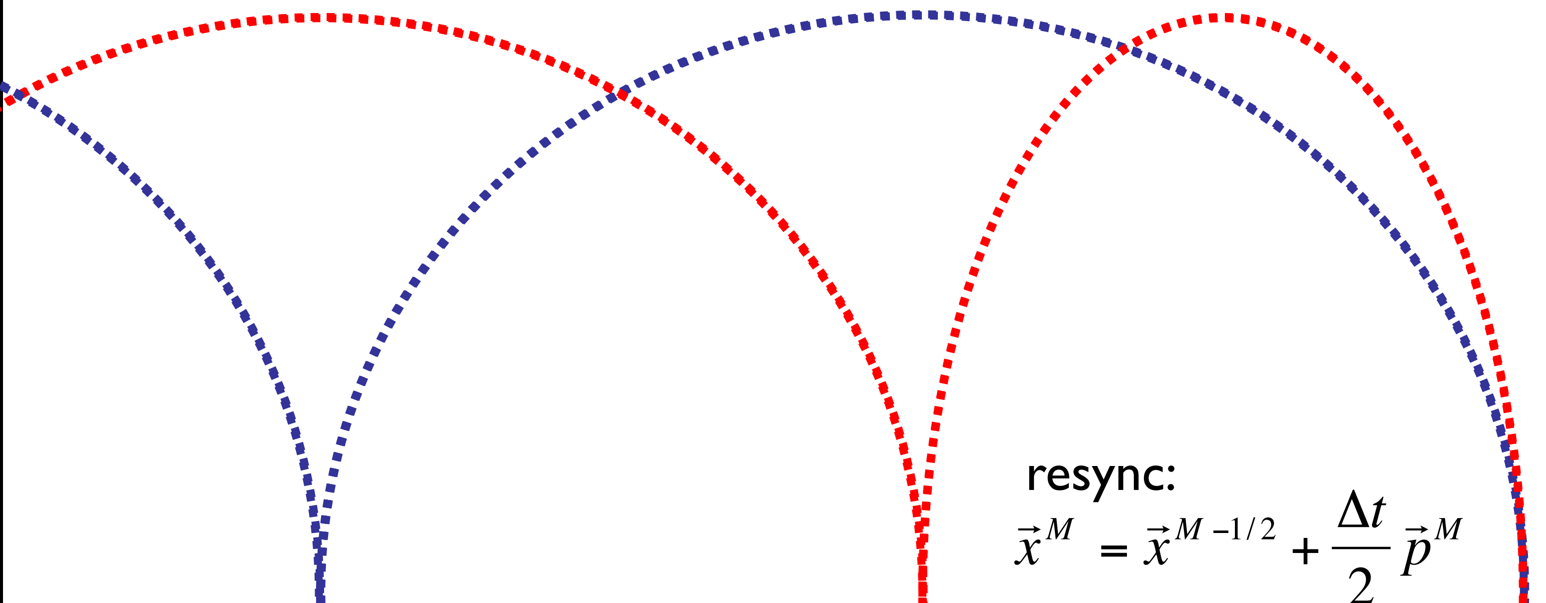


jumpstart:

$$\vec{x}^{\,1/2} = \vec{x}^{\,0} + \frac{\Delta t}{2}\,\vec{p}^{\,0}$$

- recovering the leap-frog scheme

$$\vec{x}^{\,n} \quad \vec{x}^{\,n+1}$$
$$\vec{p}^{\,n} \quad \vec{p}^{\,n+1}$$



resync:

$$\vec{x}^{\,M} = \vec{x}^{\,M-1/2} + \frac{\Delta t}{2}\,\vec{p}^{\,M}$$

- leap-frog scheme for minimal memory usage & minimal flops
  - requires only one force evaluation per time step
  - only one copy of variables stored

- Drift-Kick-Drift scheme for memory economy & synchronisation

=> • both schemes are 2nd order accurate in time

• even though DKD scheme requires N more operations,
  it is favourable for adaptive mesh refinement codes…

- leap-frog scheme for minimal memory usage & minimal flops

    - requires only one force evaluation per time step
    - only one copy of variables stored

- Drift-Kick-Drift scheme for memory economy & synchronisation

=> • both schemes are 2nd order accurate in time

• even though DKD scheme requires N more operations, it is favourable for adaptive mesh refinement codes…

$\rightarrow$ how to check your integrator?

$\rightarrow$ how to choose the correct time step?

$\rightarrow$ how to monitor the accuracy?

- the basics
- solving differential equations
- symplectic integration
- **accuracy checks**

- the time integration

$$\vec{x}_n = \vec{x} + \vec{e}_n$$

$\vec{x}_n$: numerically obtained value (at end of simulation!)

$\vec{x}$: true value

$\vec{e}_n$: global error due to numerics

- the time integration

$$\vec{x}_n = \vec{x} + \vec{e}_n$$

$\vec{x}_n$ → numerically obtained value (at end of simulation!)

$\vec{x}$ → true value

$\vec{e}_n$ → global error due to numerics

- M^th order accurate integration scheme:

$$\vec{e}_n = C(\Delta t_n)^M$$

integrate problem with different time steps...

- the time integration

$$\vec{x}_n = \vec{x} + \vec{e}_n$$

numerically obtained value
(at end of simulation!)

true value

global error due to numerics

- M^th order accurate integration scheme:

$$\vec{e}_n = C(\Delta t_n)^M$$

$$\begin{aligned}\vec{x}_n - \vec{x}_m &= \vec{x} + \vec{e}_n - (\vec{x} + \vec{e}_m)\\ &= \vec{e}_n - \vec{e}_m\\ &= C\left(\left(\Delta t_n\right)^M - \left(\Delta t_m\right)^M\right)\end{aligned}$$

$$\begin{aligned}\vec{x}_m - \vec{x}_l &= \vec{x} + \vec{e}_m - (\vec{x} + \vec{e}_l)\\ &= \vec{e}_m - \vec{e}_l\\ &= C\left(\left(\Delta t_m\right)^M - \left(\Delta t_l\right)^M\right)\end{aligned}$$

- the time integration

$$\vec{x}_n = \vec{x} + \vec{e}_n$$

numerically obtained value (at end of simulation!)

true value

global error due to numerics

- Mth order accurate integration scheme:

$$\vec{e}_n = C(\Delta t_n)^M$$

$$\begin{aligned}\vec{x}_n - \vec{x}_m &= \vec{x} + \vec{e}_n - (\vec{x} + \vec{e}_m) \\ &= \vec{e}_n - \vec{e}_m \\ &= C\left(\left(\Delta t_n\right)^M - \left(\Delta t_m\right)^M\right)\end{aligned}$$

$$\begin{aligned}\vec{x}_m - \vec{x}_l &= \vec{x} + \vec{e}_m - (\vec{x} + \vec{e}_l) \\ &= \vec{e}_m - \vec{e}_l \\ &= C\left(\left(\Delta t_m\right)^M - \left(\Delta t_l\right)^M\right)\end{aligned}$$

$$\text{let:} \quad \Delta t_l = k\Delta t_m = \ldots = k^M \Delta t_n$$

- the time integration

$$\vec{x}_n = \vec{x} + \vec{e}_n$$

numerically obtained value
(at end of simulation!)

true value

global error due to numerics

• M^th order accurate integration scheme:

$$\vec{e}_n = C(\Delta t_n)^M$$

$$\frac{\vec{x}_n - \vec{x}_m}{\vec{x}_m - \vec{x}_l} = \ldots = \frac{1}{k^M}$$

$$\text{let :} \quad \Delta t_l = k\Delta t_m = \ldots = k^M \Delta t_n$$

- the time integration

$$\vec{x}_n = \vec{x} + \vec{e}_n$$

numerically obtained value (at end of simulation!) | true value | global error due to numerics

• second order accurate leap-frog – test in practice:

– integrate with three different choices for (constant!) time step:

$$\Delta t,\ 2\Delta t,\ \text{and}\ 4\Delta t$$

– calculate

$$\frac{\vec{x}_{\Delta t} - \vec{x}_{2\Delta t}}{\vec{x}_{2\Delta t} - \vec{x}_{4\Delta t}} \stackrel{?}{=} \frac{1}{4}$$

– repeat exercise for $(2\Delta t, 4\Delta t, \text{and } 8\Delta t)$, $(4\Delta t, 8\Delta t, \text{and } 16\Delta t)$, …

- time step criteria

  - system criterion

$$\Delta t \leq t_{dyn}$$

the time step should be smaller than the dynamical time of your system

  - acceleration/velocity criterion

$$\Delta t \leq \sqrt{\frac{\varepsilon}{a_{\max}}} \qquad \Delta t \leq \frac{\varepsilon}{v_{\max}}$$

"objects" in your system should not move farther than some preselected threshold ε

ε of order the force resolution

- momentum conservation

$$\left| \sum_{i=1}^{N} \vec{F}_i \right| = 0$$

=> development of net momentum during simulation?

• practical test:

$$\frac{\left| \sum_{i=1}^{N} \vec{F}_i \right|}{\sum_{i=1}^{N} \left| \vec{F}_i \right|} \approx 10^{-4}$$

- energy conservation

  - Hamiltonian (= total energy)

$$\mathcal{H} = \frac{1}{2m} p^2 + m\Phi(\vec{x}) \qquad \Delta\Phi = 4\pi G\rho$$


PLEASE!!!
KNOW NUMERICAL ORDER!!
1.2.3.4.5....
NOT
3.2.4.1.5....