Numerical Mathematics
Differential Equations

Pieter Collins

Department of Knowledge Engineering
Maastricht University
pieter.collins@maastrichtuniversity.nl

KEN1540, Block 5, April-May 2021

Introduction to Differential Equations **2**
Differential equations 3
Solutions of differential equations 4
Analytical solution of differential equations 5

Initial-Value Problems **6**
Initial-value problems 7
ODEs in Matlab 9

Taylor Methods **10**
Euler's method 11
Taylor methods 18

Runge-Kutta Methods **23**
Midpoint method 24
Modified Euler method 29
Second-order methods 30
Third-order methods 33
Fourth-order methods 35
Comparisons 43

Multistep Methods **44**
Multistep methods 45
Adams-Bashforth methods 46
Bootstrapping 48

Implicit methods **51**
Adams-Moulton methods 53
Predictor-corrector 55
Fully-implicit 59
Gear's method 60

Stability **63**
Stability analysis 65
Explicit methods 66

Explicit vs implicit . . . 70
Stiff systems . . . 71

Multivariable systems **73**
Multivariable systems . . . 74
Second-order systems . . . 76

Adaptive methods **79**
Adaptive methods . . . 80
Bogacki-Shampine . . . 82
Adaptive multistep . . . 86

Global errors **87**
Growth of errors . . . 88

Differential equations

A *differential equation* for a quantity y is a formula giving the time-derivative of y in terms of y itself: $\dot{y} = f(t, y)$.

If y is finite-dimensional, then we have an *ordinary differential equation (ODE)*. Other kinds of differential equation are *partial differential equation*, *stochastic differential equation*.

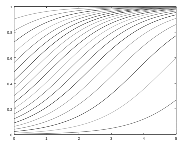
Examples

- Exponential growth $\dot{y} = ky$ or decay $\dot{y} = -ky$.
- Logistic growth $\dot{y} = ky(y_{\max} - y)$.
- Decay and forcing $\dot{y} = -ky + a\sin(\omega t)$
- Linear (first order): $\dot{y} + p(t)y = q(t)$
- Damped oscillations $\ddot{x} + \delta\dot{x} + kx = 0$.
  Equivalently: $\dot{x} = v$, $\dot{v} = \ddot{x} = -\delta v - kx$.

Solutions of differential equations

Solution curves Differential equations have multiple solutions!

- If $\frac{dy}{dt} = ky$, then any function $y(t) = Ce^{kt}$ is a solution!
  Check by substitution: $\frac{dy}{dt} = Cke^{kt} = KCe^{kt} = ky$.
- If $\dot{y} = y(1-y)$, then any function $y(t) = 1/(Ce^{-t}+1)$ is a solution.



Theorem If $\dot{y} = f(t, y)$ where f is differentiable in y, then there is a unique solution curve through any point.

- Hence solution curves do not cross each other or merge.

Analytical solution of differential equations

For some differential equations, we can compute a solution analytically.

Linear first-order equations $\dot{y}(t) + p(t)y(t) = q(t)$.

Multiply through by $e^{\int p(t)dt}$. Then by the product and chain rules

$$\frac{d}{dt}\big(e^{\int p(t)dt}y(t)\big) = e^{p(t)dt}\big(\dot{y}(t) + p(t)y(t)\big) = e^{\int p(t)dt}q(t)$$

Hence the general solution is

$$y(t) = e^{-\int p(t)dt}\big(C + \int e^{\int p(t)dt}\, q(t)\, dt\big).$$

Analytical solutions do not exist except in special cases! Even when they do exist, they are hard to calculate!

Instead, numerical methods are generally-applicable and usually very reliable!

Initial-value problems

Initial-Value Problem Find a function $y : [t_0, t_f] \to \mathbb{R}$ satisfying
$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0.$$
We say t_0 is the *initial time*, y_0 is the *initial state*, and t_f is the *final time*.

Common (but not necessary) to take $t_0 = 0$.

Numerical solution Approximate y at points $t_0 < t_1 < \cdots < t_N = t_f$ as
$$y(t_i) \approx w_i, \quad i = 0, \ldots, N.$$

Step size The *step sizes* are $h_i = t_{i+1} - t_i$, so $t_{i+1} = t_i + h_i$.

Simplest is to take constant step size $h_i = h$.

Then $h = (t_f - t_0)/N$ and $t_i = t_0 + ih$.

Chemical reactions

Example Reactions $A \longrightarrow Y$ and $Y + Y \longrightarrow B$.

The concentrations $a = [A]$ and $y = [Y]$ satisfy the differential equations
$$\frac{da}{dt} = -k_1 a; \qquad \frac{dy}{dt} = k_1 a - k_2 y^2$$
where k_1, k_2 are rate constants.

Since a decays exponentially as $a = a_0 e^{-k_1 t}$ where $a_0 = [A(0)]$, we have
$$\frac{dy}{dt} = a_0 k_1 e^{-k_1 t} - k_2 y^2.$$
Use initial condition $t_0 = 0$ and $y_0 = [Y(0)] = 0$.

Henceforth, take constants $a_0 = k_1 = k_2 = 1$ (in appropriate units) so
$$\frac{dy}{dt} = e^{-t} - y^2.$$

Solution of ODEs in Matlab

Example Solve the following initial value problem in Matlab.

$$\frac{dy}{dt} = a_0 k_1 e^{-k_1 t} - k_2 y^2; \quad y(0) = 0$$

Set up the differential equation:

```
a0=1; k1=1; k2=1;
f = @(t,y) a0*k1*exp(-k1*t) - k2*y.^2;
t0=0; tf=5;
y0=0;
```

Solve using a built-in solver e.g. `ode45`:

```
[ts,ys]=ode45(f,[t0,tf],y0)
```

Other solvers are available, namely `ode23`, `ode23s`, `ode15s`.

Plot the solution:

```
plot(ts,ys)
```


Taylor Methods

Euler's method

First-order approximation The differential equation

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

is approximated by

$$\frac{\delta y}{\delta t} \approx f(t,y).$$

Then

$$y(t+\delta t) - y(t) = \delta y \approx f(t,y)\,\delta t,$$

yielding the time-step of *Euler's method*

$$y(t+\delta t) \approx y(t) + f(t,y(t))\delta t.$$

Alternatively, write

$$y(t+h) \approx y(t) + hf(t,y(t)).$$

Euler's method Compute $w_i \approx y(t_i)$ by $w_0 = y(t_0)$ and

$$w_{i+1} = w_i + h_i f(t_i, w_i)$$

where $h_i = t_{i+1} - t_i$ is the step size.

Euler's method

Example Solve $dy/dt = f(t, y) := e^{-t} - y^2$ with $y(0) = 0$ up to time 1.

Initialise $t_0 = 0$ and $w_0 = y_0 = y(0) = 0$.
Take $t_f = 1$, and aim to approximate $y(1)$ i.e. $y(t)$ when $t = 1$.

Try a single step, $h = 1.0$:
Formula $w_1 = w_0 + hf(t_0, w_0)$.
Evaluate $f(t_0, w_0) = (e^{-t_0} - w_0^2) = e^{-0} - 0^2 = 1$, so

$$y(1.0) \approx w_1 = w_0 + hf(t_0, w_0) = 0.0000 + 1.0 \times 1.0000 = 1.0000.$$

The exact value is $y(1) = 0.503347\,(6\,\text{sf})$.
Absolute error ≈ 0.50.
Relative error

$$|w_1 - y(1)|/|y(1)| = 0.4967/0.503347 \approx 99\%!$$

Very poor approximation...

Euler's method

Example Solve $dy/dt = f(t, y) := e^{-t} - y^2$ with $y(0) = 0$ up to time 1.

Take $n = 2$ steps, each with step size $h = 0.5$.

Times $t_0 = 0$, $t_1 = t_0 + h = 0.5$, $t_2 = t_1 + h = 1.0$.

Initialise $w_0 = y(t_0) = y(0) = 0.000$. Solution is $y(1) = y(t_2) \approx w_2$

$$\begin{aligned} y(0.5) \approx w_1 &= w_0 + hf(t_0, w_0) \\ &= 0.000 + 0.5 \times f(0.0, 0.000) \\ &= 0.0000 + 0.5 \times 1.000 = 0.500. \\ y(1.0) \approx w_2 &= w_1 + hf(t_1, w_1) \\ &= 0.500 + 0.5 \times f(0.5, 0.500) \\ &= 0.5000 + 0.5 \times 0.357 = 0.678. \end{aligned}$$

$$\begin{aligned} f(t_0, w_0) &= f(0.0, 0.000) \\ &= e^{-0.0} - 0.000^2 \\ &= 1.000 \\ f(t_1, w_1) &= f(0.5, 0.500) \\ &= e^{-0.5} - 0.500^2 \\ &= 0.607 - 0.250 \\ &= 0.357 \end{aligned}$$

Better approximation, absolute error ≈ 0.17; relative error 35%.

Euler's method

Example Use fixed step size $h = 0.2$:

i	t_i	w_i	$f(t_i, w_i)$	$w_i + hf(t_i, w_i)$
0	0.00000	0.00000	1.00000	0.20000
1	0.20000	0.20000	0.77873	0.35575
2	0.40000	0.35575	0.54376	0.46450
3	0.60000	0.46450	0.33305	0.53111
4	0.80000	0.53111	0.16725	0.56456
5	1.00000	0.56456		

Better approximation; absolute error $|w_5 - y(1.0)| \approx 0.061$, relative error 12%.

With step size $h = 0.1$, find $y(1.0) \approx w_{10} = 0.5329$, relative error 5.8%.

Error approximately halves on halving the step-size: first order method!

Error of Euler's method (Non-examinable)

Error estimate By Taylor's theorem, $y(t+h) = y(t) + h\dot{y}(t) + \frac{1}{2}h^2\ddot{y}(\tau)$.

The first derivative of y is given by the differential equation $\dot{y}(t) = f(t, y(t))$.

The second derivative of y is given by

$$\ddot{y}(t) = \frac{d}{dt}\dot{y}(t) = \frac{d}{dt}f(t,y(t)) = \frac{\partial f(t,y)}{\partial t} + \frac{\partial f(t,y)}{\partial y}\frac{dy(t)}{dt}$$
$$= \frac{\partial f(t,y)}{\partial t} + \frac{\partial f(t,y)}{\partial y}\, f(t,y)$$

Using notation $f_{,t} = \partial f/\partial t$ and $f_{,y} = \partial f/\partial y$ we have

$$\ddot{y}(t) = f_{,t}(t,y(t)) + f_{,y}(t,y(t))\, f(t,y(t)).$$

Obtain local error estimate

$$\Big|y(t_1) - w_1\Big| = \Big|\big(\cancel{y(t_0)} + \cancel{h_0\dot{y}(t_0)} + \tfrac{1}{2}h_0^2\ddot{y}(\tau)\big) - \big(\cancel{w_0} + \cancel{h_0 f(t_0,w_0)}\big)\Big|$$
$$= \Big|\tfrac{1}{2}h_0^2\ddot{y}(\tau)\Big| = \frac{h_0^2}{2}\Big|f_{,t}(\tau,y(\tau)) + f_{,y}(\tau,y(\tau))f(\tau,y(\tau))\Big|$$
$$= \frac{h_0^2}{2}\Big|f_{,t}(\tau,\eta) + f_{,y}(\tau,\eta)f(\tau,\eta)\Big| \quad \text{for some } \tau,\eta.$$

Euler's method

Theorem (Local error bound for Euler's method) Let y be the solution of the differential equation $dy/dt = f(t,y)$, and let

$$w_{i+1} = y(t_i) + h_i f(t_i, y(t_i))$$

be given by Euler's method.

Then there exists $\tau \in [t_i, t_{i+1}]$ such that

$$|w_{i+1} - y(t_{i+1})| = \frac{1}{2}\big|f_{,t}(\tau,y(\tau)) + f_{,y}(\tau,y(\tau))\, f(\tau,y(\tau))\big|h_i^2.$$

In particular, if $y(t) \in [y_{\min}, y_{\max}]$ for all $t \in [t_i, t_{i+1}]$, and

$$\big|f_{,t}(\tau,\eta) + f_{,y}(\tau,\eta)\, f(\tau,\eta)\big| \le M$$

for all $\tau \in [t_i, t_{i+1}]$ and $\eta \in [y_{\min}, y_{\max}]$, then

$$|w_{i+1} - y(t_{i+1})| \le \frac{M}{2}h_i^2 = O(h_i^2).$$

Euler's method

Growth of global errors Need to approximate y at some *fixed* time t_f.

For steps size h, require $N = (t_f - t_0)/h$ steps.

For $i \geq 1$, $y(t_i) \neq w_i$, so error $w_{i+1} - y(t_{i+1})$ contains errors *accumulated* from previous time steps.

If $f_{,y}(\tau, \eta) \leq L$, then error e_i at time t_i satisfies $|e_{i+1}| \leq e^{-h_i L}|e_i|$ *if the step is exact.*

Theorem (Error bound for Euler's method) If

$$f_{,y}(\tau,\eta) \leq L \text{ and } \big|f_{,t}(\tau,\eta) + f_{,y}(\tau,\eta)\, f(\tau,\eta)\big| \leq M$$

for all $\tau \in [0, T]$ and $\eta \in [y_{\min}, y_{\max}]$, then for a fixed step size $h = T/N$ and for all $i \leq N$,

$$|w_i - y(t_i)| \leq h\, \frac{M}{2}\, \frac{\exp(LT) - 1}{L}\,.$$

Although *local* (single-step) error is $O(h^2)$, *global* error (up to a fixed time) is $O(h)$, since we require $O(1/h)$ steps.

Second-order Taylor method (Non-examinable)

Second-order Taylor method Use second-order term in Taylor expansion to improve Euler's method.

$$y(t+h) \approx y(t) + hf(t,y) + \tfrac{1}{2}h^2\big(f_{,t}(t,y) + f_{,y}(t,y)f(t,y)\big),$$

so take

$$w_{i+1} = w_i + h_i f(t_i, w_i) + \tfrac{1}{2}h_i^2\big(f_{,t}(t_i,w_i) + f_{,y}(t_i,w_i)f(t_i,w_i)\big).$$

Second-order Taylor method (Non-examinable)

Example Let $f(t,y) = e^{-t} - y^2$, so $f_{,t}(t,y) = -e^{-t}$ and $f_{,y}(t,y) = -2y$.
Initial condition $y(0) = 0$.

For step size $h = 1.0$,

$$f(0.0, 0.000) = 1.000, \quad f_{,t}(0.0, 0.000) = -1.000, \quad f_{,y}(0.0, 0.000) = 0.000;$$

$$\begin{aligned} y(1.0) \approx w_1 &= w_0 + hf(t_0,w_0) + \tfrac{1}{2}h^2\big(f_{,t}(t_0,w_0) + f_{,y}(t_0,w_0)f(t_0,w_0)\big) \\ &= 0.0000 + 1.0 \times 1.0000 + \tfrac{1}{2} \times 1.0^2 \times (-1.0000 + 0.0000 \times 1.0000) \\ &= 0.0000 + 1.0000 - 0.5000 = 0.5000 \end{aligned}$$

Relative error 0.7%, much better!

Second-order Taylor method (Non-examinable)

Example Let $f(t,y)=e^{-t}-y^2$, so $f_{,t}(t,y)=-e^{-t}$ and $f_{,y}(t,y)=-2y$.
Initial condition $y(0)=0$.

For step size $h=0.5$,

$$\begin{aligned}y(0.5)\approx w_1 &= w_0+hf(t_0,w_0)+\tfrac{1}{2}h^2\big(f_{,t}(t_0,w_0)+f_{,y}(t_0,w_0)f(t_0,w_0)\big)\\ &= 0.0000+0.5\times 1.0000+\tfrac{1}{2}\times 0.5^2\times\big(-1.0000+0.0000\times 1.0000\big)\\ &= 0.3750.\\ y(1.0)\approx w_2 &= w_1+hf(t_1,w_1)+\tfrac{1}{2}h^2\big(f_{,t}(t_1,w_1)+f_{,y}(t_1,w_1)f(t_1,w_1)\big)\\ &= 0.3750+0.5\times 0.4659+\tfrac{1}{2}\times 0.5^2\times\big(-0.6065-0.7500\times 0.4659\big)\\ &= 0.4885.\end{aligned}$$

Relative error 3%, still much better than Euler's method.

For $h=0.2$, $y(1.0)\approx w_5=0.500708$, with relative error 0.52%.
For $h=0.1$, $y(1.0)\approx w_{10}=0.502675$, with relative error 0.13%.

Halving the step size multiplies the error by a quarter, so method is *second* order; global error $O(h^2)$.

Higher-order Taylor methods (Non-examinable)

Higher-order derivatives Consider (for simplicity) the *autonomous* differential equation $\dot{y}=f(y)$.

Then

$$\begin{aligned}\dot{y} &= f(y);\\ \ddot{y} &= f'(y)f(y);\\ \dddot{y} &= f''(y)f(y)^2+f'(y)^2f(y);\\ \ddddot{y} &= f'''(y)f(y)^3+4f''(y)f'(y)f(y)^2+f'(y)^3f(y);\\ \overset{\cdot\cdot\cdot\cdot\cdot}{y} &= f''''(y)f(y)^4+7f'''(y)f'(y)f(y)^3+4f''(y)^2f(y)^3\\ &\qquad +11f''(y)f'(y)^2f(y)^2+f'(y)^4f(y).\end{aligned}$$

Higher-order Taylor methods Higher order Taylor methods can be constructed from these derivatives. e.g. third-order

$$w_{i+1}=w_i+h_if(w_i)+\frac{h_i^2}{2}f'(w_i)f(w_i)+\frac{h_i^3}{6}\big(f''(w_i)f(w_i)^2+f'(w_i)^2f(w_i)\big).$$

Higher-order Taylor methods (Non-examinable)

Advantages and disadvantages of Taylor methods

Taylor methods require the computation of partial derivatives, which is inconvenient.

Taylor methods can be used to give *rigorous* bounds on local and global errors.

e.g. For the second-order Taylor method applied to $\dot{y}=f(y)$, obtain local error

$$\begin{aligned}|w_1-y(t_1)| &\leq \frac{h^3}{6}\sup_{\tau\in[t_0,t_1]}\big|\dddot{y}(\tau)\big|\\ &\leq \frac{h^3}{6}\sup_{\eta\in[y_{\min},y_{\max}]}\big|f''(\eta)f(\eta)^2+f'(\eta)^2f(\eta)\big|,\end{aligned}$$

and global error $O(h^2)$.

Midpoint method

Midpoint method Expect derivative $\dot{y}(t+\frac{h}{2})$ at midpoint of interval $[t,t+h]$ to give a better approximation to the average change $\frac{1}{h}(y(t+h)-y(t))$ than derivative $\dot{y}(t)$ at left endpoint.

Can show

$$y(t+h)=y(t)+h\,\dot{y}(t)+O(h^2); \qquad y(t+h)=y(t)+h\,\dot{y}(t+\tfrac{h}{2})+O(h^3).$$

Would like to use approximation

$$y(t+h)\approx y(t)+hf(t+\tfrac{h}{2},y(t+\tfrac{h}{2})).$$

Problem: Don't know $y(t+\frac{h}{2})$.

Solution: Estimate $y(t+\frac{h}{2})$ by Euler's method!

$$y(t+\tfrac{h}{2})\approx y(t)+\tfrac{h}{2}f(t,y(t))$$

Then

$$y(t+h)\approx y(t)+hf\big(t+\tfrac{h}{2},y(t)+\tfrac{h}{2}f(t,y(t))\big).$$

Obtain midpoint method:

$$w_{i+1}=w_i+h_if(t_i+\tfrac{1}{2}h_i,\ w_i+\tfrac{1}{2}h_if(t_i,w_i)).$$

Midpoint method

Example Solve $\dot{y}=e^{-t}-y^2$ with $y(0)=0$ up to $t=1$ using $h=0.5$.

Times $t_0=0.0, t_1=0.5, t_2=1.0$. $w_0=y(t_0)=0.000$. Solution $y(1)=y(t_2)\approx w_2$.

$$\begin{aligned} w_1 &= w_0+hf(t_0+\tfrac{1}{2}h,\ w_0+\tfrac{1}{2}hf(t_0,w_0))\\ &= w_0+hf(t_0+\tfrac{1}{2}h,\ w_0+\tfrac{1}{2}hf(0.0,0.0000))\\ &= w_0+hf(0.0+\tfrac{1}{2}\times 0.5,\ 0.0000+\tfrac{1}{2}\times 0.5\times 1.0000)\\ &= w_0+hf(0.25,0.2500)=0.0000+0.5\times 0.7163\\ &= 0.3582 \end{aligned}$$

$$\begin{aligned} w_2 &= w_1+hf(t_1+\tfrac{1}{2}h,\ w_1+\tfrac{1}{2}hf(t_1,w_1))\\ &= w_1+hf(t_1+\tfrac{1}{2}h,\ w_1+\tfrac{1}{2}hf(0.5,0.3582))\\ &= w_1+hf(0.5+\tfrac{1}{2}\times 0.5,\ 0.3582+\tfrac{1}{2}\times 0.5\times 0.4783)\\ &= w_1+hf(0.75,0.4777)=0.3582+0.5\times 0.2442\\ &= 0.4802. \end{aligned}$$

$$\begin{aligned} f(t_0,w_0) &= f(0.0,0.0000)\\ &= 1.0000\\ f(t_0+\tfrac{1}{2}h,w_0+\tfrac{1}{2}hf(t_0,w_0)) &= f(0.25,0.2500)\\ &= e^{-0.2500}-0.2500^2\\ &= 0.7788-0.0625\\ &= 0.7163\\ f(t_1,w_1) &= f(0.5,0.3582)\\ &= 0.4783\\ f(t_1+\tfrac{1}{2}h,w_1+\tfrac{1}{2}hf(t_1,w_1)) &= f(0.75,0.4777)\\ &= 0.2442 \end{aligned}$$

Exact value

$y(1)=0.503347$ (6 sf).

Relative error 4.5%, better than 12% for Euler's method with $h=0.2$.

Midpoint method

Example For $\dot{y} = e^{-t} - y^2$, $y(0) = 0$, take step-size $h = 0.1$.

Work to machine precision, display w_i to 4 dp:

$$\begin{aligned} w_1 &= w_0 + hf(t_0 + \tfrac{1}{2}h,\ w_0 + \tfrac{1}{2}hf(t_0, w_0)) \\ &= w_0 + hf(t_0 + \tfrac{1}{2}h,\ w_0 + \tfrac{1}{2}hf(0.0, 0.00000)) \\ &= w_0 + hf(0.0 + \tfrac{1}{2}\times 0.1,\ 0.00000 + \tfrac{1}{2}\times 0.1 \times 1.0000) \\ &= w_0 + hf(0.05, 0.05000) \\ &= 0.0000 + 0.1 \times 0.9487 = 0.09487 \\ w_2 &= w_1 + hf(t_1 + \tfrac{1}{2}h,\ w_1 + \tfrac{1}{2}hf(t_1, w_1)) \\ &= w_1 + hf(t_1 + \tfrac{1}{2}h,\ w_1 + \tfrac{1}{2}hf(0.1, 0.09487)) \\ &= w_1 + hf(0.1 + \tfrac{1}{2}\times 0.1,\ 0.09487 + \tfrac{1}{2}\times 0.1 \times 0.89584) \\ &= w_1 + hf(0.15, 0.13966) \\ &= 0.09487 + 0.1 \times 0.8412 = 0.17899. \end{aligned}$$

Midpoint method

Example For $\dot{y} = e^{-t} - y^2$, $y(0) = 0$, take step-size $h = 0.1$.
Let $t_{i+\frac{1}{2}} = t_i + \frac{1}{2}h$, $\hat{w}_{i+\frac{1}{2}} = w_i + \frac{1}{2}hf(t_i, w_i)$, $w_{i+1} = w_i + hf(t_{i+\frac{1}{2}}, \hat{w}_{i+\frac{1}{2}})$.
Work to machine precision, display w_i to 5 decimal places:

i	t_i	w_i	$f(t_i, w_i)$	$t_{i+\frac{1}{2}}$	$\hat{w}_{i+\frac{1}{2}}$	$f(t_{i+\frac{1}{2}}, \hat{w}_{i+\frac{1}{2}})$	w_{i+1}
0	0.0	0.00000	1.0000	0.05	0.05000	0.9487	0.09487
1	0.1	0.09487	0.8958	0.15	0.13966	0.8412	0.17899
2	0.2	0.17899	0.7867	0.25	0.21833	0.7311	0.25211
3	0.3	0.25211	0.6773	0.35	0.28597	0.6229	0.31440
4	0.4	0.31440	0.5715	0.45	0.34297	0.5200	0.36640
5	0.5	0.36640	0.4723	0.55	0.39001	0.4248	0.40888
6	0.6	0.40888	0.3816	0.65	0.42796	0.3389	0.44277
7	0.7	0.44277	0.3005	0.75	0.45780	0.2628	0.46905
8	0.8	0.46905	0.2293	0.85	0.48052	0.1965	0.48870
9	0.9	0.48870	0.1677	0.95	0.49709	0.1396	0.50267
10	1.0	0.50267					

Exact value $y(1) = 0.503347$; relative error 0.13%.

Midpoint method

Example For $\dot{y} = e^{-t} - y^2$, $y(0) = 0$, approximating $y(1) = 0.503347$ (6 sf) with different step-sizes gives

n	1	2	5	10
h	1.0	0.5	0.2	0.1
$y(1.0) \approx w_n$	0.356531	0.480228	0.500418	0.502666
error	29%	4.6%	0.58%	0.14%

The midpoint method is *second-order*, with global errors $O(h^2)$.

Modified Euler (trapezoid) method

Modified Euler (trapezoid) method Trapezoid rule gives

$$y(t+h) = y(t) + \tfrac{h}{2}\big(\dot y(t) + \dot y(t+h)\big) + O(h^3).$$

Estimate $y(t+h)$ by Euler's method, $y(t+h) \approx y(t) + hf(t, y(t))$.
Then

$$y(t+h) \approx y(t) + \tfrac{1}{2}h\big(f(t,y(t)) + f(t{+}h, y(t){+}hf(t,y(t)))\big).$$

Obtain *modified Euler* (trapezoid) method:

$$w_{i+1} = w_i + \tfrac{1}{2}h_i\big(f(t_i,w_i) + f(t_i{+}h_i, w_i{+}h_i f(t_i,w_i))\big).$$

Example For $\dot y = e^{-t} - y^2$, $y(0) = 0$, approximating $y(1) = 0.503347$ (6 sf) with different step-sizes gives

n	1	2	5	10
h	1.0	0.5	0.2	0.1
w_n	0.183940	0.468458	0.499972	0.502639
error	63%	6.9%	0.67%	0.14%

Second-order Runge-Kutta methods

General second-order Runge-Kutta method

$$y(t+h) \approx y(t) + h\big(a_1 f(t,y(t)) + a_2 f(t+\alpha h, y(t) + \beta h f(t,y(t)))\big)$$

Differentiate with respect to h:

$$\dot y(t+h) = \big(a_1 f(t,y(t)) + a_2 f(t+\alpha h, y(t) + \beta h f(t,y(t)))\big) \\ + h a_2\big(\alpha f_{,t}(t+\alpha h, y+\beta h f(t,y)) + \beta f_{,y}(t+\alpha h, y+\beta h f(t,y)) f(t,y)\big)$$

Hence

$$\dot y(t) = a_1 f(t,y) + a_2 f(t,y) = (a_1+a_2) f(t,y)$$
$$\ddot y(t) = 2a_2\big(\alpha f_{,t}(t,y) + \beta f_{,y}(t,y) f(t,y)\big)$$

From Taylor's theorem,

$$\dot y(t) = f(t,y); \quad \ddot y(t) = f_{,t}(t,y) + f_{,y}(t,y) f(t,y).$$

Comparing coefficients gives $a_1 + a_2 = 1$ and $2a_2\alpha = 2a_2\beta = 1$, so

$$\beta = \alpha \quad \text{and} \quad a_1 = 1{-}1/2\alpha,\ a_2 = 1/2\alpha.$$

Obtain method

$$w_{i+1} = w_i + h_i\big((1-\tfrac{1}{2\alpha}) f(t_i,w_i) + \tfrac{1}{2\alpha} f(t_i+\alpha h_i, w_i + \alpha h_i f(t_i,w_i))\big)$$

Ralston's method

General second-order Runge-Kutta method

$$w_{i+1} = w_i + h_i\big((1 - \tfrac{1}{2\alpha})f(t_i, w_i) + \tfrac{1}{2\alpha}f(t_i + \alpha h_i, w_i + \alpha h_i f(t_i, w_i))\big)$$

Midpoint and modified Euler methods

Taking $\alpha = \frac{1}{2}$ gives the midpoint method with $a_1 = 0$, $a_2 = 1$.
Taking $\alpha = 1$ gives the modified Euler method with $a_1 = a_2 = \frac{1}{2}$.

Ralston's Method

Taking $\alpha = \frac{2}{3}$ gives $a_2 = \frac{3}{4}$, $a_1 = \frac{1}{4}$ and

$$w_{i+1} = w_i + \tfrac{1}{4}h_i\big(f(t_i, w_i) + 3f(t_i + \tfrac{2}{3}h_i, w_i + \tfrac{2}{3}h_i f(t_i, w_i))\big)$$

This can be written in a standard form for Runge-Kutta methods:

$$\begin{aligned} k_{i,1} &= h_i f(t_i, w_i); \\ k_{i,2} &= h_i f(t_i + \tfrac{2}{3}h_i, w_i + \tfrac{2}{3}k_{i,1}); \\ w_{i+1} &= w_i + \tfrac{1}{4}\big(k_{i,1} + 3k_{i,2}\big) = w_i + \tfrac{1}{4}k_{i,1} + \tfrac{3}{4}k_{i,2} \\ &= w_i + k_i \text{ where } k_i = \tfrac{1}{4}k_{i,1} + \tfrac{3}{4}k_{i,2}. \end{aligned}$$

Runge-Kutta Methods

Example $\dot{y}(t) = e^{-t} - y^2$, $y(0) = 0$. Step size $h = 0.1$.

	$y(0.1)$	$y(1.0)$
Euler	0.1000000000000000	0.532904863460103
Midpoint	0.0948729424500714	0.502665926212565
Trapezoid	0.0947418709017980	0.502638707657163
Ralston	0.0948296905440380	0.502658823715687
Exact	$0.094854320\cdots$	$0.5033467\cdots$

Third-order Runge-Kutta methods

Heun's third-order method

$$\begin{aligned} k_{i,1} &= h_i f(t_i, w_i); \\ k_{i,2} &= h_i f(t_i + \tfrac{1}{3}h_i, w_i + \tfrac{1}{3}k_{i,1}); \\ k_{i,3} &= h_i f(t_i + \tfrac{2}{3}h_i, w_i + \tfrac{2}{3}k_{i,2}); \\ w_{i+1} &= w_i + \tfrac{1}{4}\big(k_{i,1} + 3k_{i,3}\big). \end{aligned}$$

Local error $O(h^4)$, global error $O(h^3)$.

Kutta's third-order method

$$\begin{aligned} k_{i,1} &= h_i f(t_i, w_i); \\ k_{i,2} &= h_i f(t_i + \tfrac{1}{2}h_i, w_i + \tfrac{1}{2}k_{i,1}); \\ k_{i,3} &= h_i f(t_i + h_i, w_i - k_{i,1} + 2k_{i,2}); \\ w_{i+1} &= w_i + \tfrac{1}{6}\big(k_{i,1} + 4k_{i,2} + k_{i,3}\big). \end{aligned}$$

Third-order Runge-Kutta methods

General third-order method (Non-examinable)

$$\begin{aligned} k_{i,1} &= h_i f(t_i, w_i); \\ k_{i,2} &= h_i f(t_i + c_2 h_i, w_i + a_{21} k_{i,1}); \\ k_{i,3} &= h_i f(t_i + c_3 h_i, w_i + a_{31} k_{i,1} + a_{32} k_{i,2}); \\ w_{i+1} &= w_i + \big(b_1 k_{i,1} + b_2 k_{i,2} + b_3 k_{i,3}\big). \end{aligned}$$

where

$$b_1 + b_2 + b_3 = 1;\ c_2 b_2 + c_3 b_2 = \tfrac{1}{2};\ c_2^2 b_2 + c_3^2 b_3 = \tfrac{1}{3};$$
$$c_2 = a_{21};\ c_3 = a_{31} + a_{32};\ c_2 b_3 a_{32} = \tfrac{1}{6}.$$

Can choose c_2, c_3 freely. Then

$$b_3 = (3c_2 - 2)/6c_3(c_2 - c_3),\ b_2 = (3c_3 - 2)/6c_2(c_3 - c_2),\ b_1 = 1 - b_2 - b_3,$$
$$a_{32} = 1/6c_2 b_3,\ a_{31} = c_3 - a_{32},\ a_{21} = c_2.$$

Classical fourth-order Runge-Kutta method

Classical fourth-order Runge-Kutta method

$$\begin{aligned} k_{i,1} &= h_i f(t_i, w_i); \\ k_{i,2} &= h_i f(t_i + \tfrac{1}{2} h_i, w_i + \tfrac{1}{2} k_{i,1}); \\ k_{i,3} &= h_i f(t_i + \tfrac{1}{2} h_i, w_i + \tfrac{1}{2} k_{i,2}); \\ k_{i,4} &= h_i f(t_i + h_i, w_i + k_{i,3}); \\ w_{i+1} &= w_i + \tfrac{1}{6}(k_{i,1} + 2k_{i,2} + 2k_{i,3} + k_{i,4}) \end{aligned}$$

Local error $O(h^5)$, global error $O(h^4)$.

May set $k_i = \frac{1}{6}(k_{i,1} + 2k_{i,2} + 2k_{i,3} + k_{i,4})$, so $w_{i+1} = w_i + k_i$.

Often drop the subscript i:

$$k_1 = hf(t, w); \qquad k_2 = hf(t + \tfrac{1}{2}h, w + \tfrac{1}{2}k_1);$$
$$k_3 = hf(t + \tfrac{1}{2}h, w + \tfrac{1}{2}k_2); \qquad k_4 = hf(t + h, w + k_3);$$
$$w' = w + \tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Fourth-order Runge-Kutta methods

Fourth-order Runge-Kutta method

$$\begin{aligned} k_{i,1} &= h_i f(t_i, w_i); \\ k_{i,2} &= h_i f(t_i + \tfrac{1}{3} h_i, w_i + \tfrac{1}{3} k_{i,1}); \\ k_{i,3} &= h_i f(t_i + \tfrac{2}{3} h_i, w_i - \tfrac{1}{3} k_{i,1} + k_{i,2}); \\ k_{i,4} &= h_i f(t_i + h_i, w_i + k_{i,1} - k_{i,2} + k_{i,3}); \\ w_{i+1} &= w_i + \tfrac{1}{8}(k_{i,1} + 3k_{i,2} + 3k_{i,3} + k_{i,4}) \end{aligned}$$

Local error $O(h^5)$, global error $O(h^4)$.

May set $k_i = \frac{1}{8}(k_{i,1} + 3k_{i,2} + 3k_{i,3} + k_{i,4})$, so $w_{i+1} = w_i + k_i$.

Fourth-order Runge-Kutta methods

Example Solve $\dot{y}(t) = e^{-t} - y^2$ with $y(0) = 0$ using $h = 0.5$.

Initialise $t_0 = 0.0,\ w_0 = y(t_0) = y(0) = 0.0000.$

First step $h_0 = h = 0.5,\ t_1 = t_0 + h_0 = 0.0 + 0.5 = 0.5.$

$$\begin{aligned}
k_{0,1} &= h_0 f(t_0, w_0) = 0.5 f(0.00, 0.0000) \\
&= 0.5 \times f(0.00, 0.0000) = 0.5 \times 1.0000 = 0.5000 \\
k_{0,2} &= h_0 f(t_0 + \tfrac{1}{2}h_0, w_0 + \tfrac{1}{2}k_{0,1}) = 0.5 f(0.0 + \tfrac{1}{2}\times 0.5, 0.0000 + \tfrac{1}{2}\times 0.5000) \\
&= 0.5 \times f(0.25, 0.2500) = 0.5 \times 0.7163 = 0.3582 \\
k_{0,3} &= h_0 f(t_0 + \tfrac{1}{2}h_0, w_0 + \tfrac{1}{2}k_{0,2}) = 0.5 f(0.0 + \tfrac{1}{2}\times 0.5, 0.0000 + \tfrac{1}{2}\times 0.3582) \\
&= 0.5 \times f(0.25, 0.1791) = 0.5 \times 0.7467 = 0.3734 \\
k_{0,4} &= h_0 f(t_0 + h_0, w_0 + k_{0,3}) = 0.5 f(0.0 + 0.5, 0.0000 + 0.3734) \\
&= 0.5 \times f(0.50, 0.3734) = 0.5 \times 0.4671 = 0.2336 \\
k_0 &= \tfrac{1}{6}(k_{0,1} + 2k_{0,2} + 2k_{0,3} + k_{0,4}) \\
&= \tfrac{1}{6}(0.5000 + 2\times 0.3582 + 2\times 0.3734 + 0.2336) = 0.3661 \\
w_1 &= w_0 + k_0 = 0.0000 + 0.3661 = 0.3661
\end{aligned}$$

Fourth-order Runge-Kutta methods

Example Solve $\dot{y}(t) = e^{-t} - y^2$ with $y(0) = 0$ using $h = 0.5$.

First step gave $t_1 = 0.5,\ w_1 = 0.3661.$

Second step $h_1 = h = 0.5,\ t_2 = t_1 + h_1 = 0.5 + 0.5 = 1.0.$

$$\begin{aligned}
k_{1,1} &= h_1 f(t_1, w_1) = 0.5 f(0.50, 0.3661) \\
&= 0.5 \times f(0.50, 0.3661) = 0.5 \times 0.4725 = 0.2363 \\
k_{1,2} &= h_1 f(t_1 + \tfrac{1}{2}h_1, w_1 + \tfrac{1}{2}k_{1,1}) = 0.5 f(0.5 + \tfrac{1}{2}\times 0.5, 0.3661 + \tfrac{1}{2}\times 0.2363) \\
&= 0.5 \times f(0.75, 0.4842) = 0.5 \times 0.2379 = 0.1189 \\
k_{1,3} &= h_1 f(t_1 + \tfrac{1}{2}h_1, w_1 + \tfrac{1}{2}k_{1,2}) = 0.5 f(0.5 + \tfrac{1}{2}\times 0.5, 0.3661 + \tfrac{1}{2}\times 0.1189) \\
&= 0.5 \times f(0.75, 0.4256) = 0.5 \times 0.2913 = 0.1456 \\
k_{1,4} &= h_1 f(t_1 + h_1, w_1 + k_{1,3}) = 0.5 f(0.5 + 0.5, 0.3661 + 0.1456) \\
&= 0.5 \times f(1.00, 0.5117) = 0.5 \times 0.1060 = 0.0530 \\
k_1 &= \tfrac{1}{6}(k_{1,1} + 2k_{1,2} + 2k_{1,3} + k_{1,4}) \\
&= \tfrac{1}{6}(0.2363 + 2\times 0.1189 + 2\times 0.1456 + 0.0530) = 0.1365 \\
y(1) \approx w_2 &= w_1 + k_1 = 0.3661 + 0.1365 = 0.5025
\end{aligned}$$

Fourth-order Runge-Kutta methods

Example $\dot{y}(t) = e^{-t} - y^2$, $y(0) = 0$. Take $h = 0.5$. Take two steps. Calulate:

i	t_i	w_i	$k_{i,1}$	$k_{i,2}$	$k_{i,3}$	$k_{i,4}$	k_i
0	0.0	0.000000	0.500000	0.358150	0.373366	0.233564	0.366100
1	0.5	0.366100	0.236251	0.118946	0.145627	0.053008	0.136401
2	1.0	0.502501					

Exact value $y(1) = 0.503347$ (6 dp); relative error 0.17%.

Fourth-order Runge-Kutta methods

Example Take $h = 0.2$.

$$\begin{aligned}
k_{0,1} &= h_0 f(t_0, w_0) \\
&= 0.2 f(0.0, 0.00000) = 0.2 \times 1.00000 = 0.20000 \\
k_{0,2} &= h_0 f(t_0 + \tfrac{1}{2}h_0, w_0 + \tfrac{1}{2}k_{0,1}) \\
&= 0.2 f(0.1, 0.10000) = 0.2 \times 0.89484 = 0.17897 \\
k_{0,3} &= h_0 f(t_0 + \tfrac{1}{2}h_0, w_0 + \tfrac{1}{2}k_{0,2}) \\
&= 0.2 f(0.1, 0.08948) = 0.2 \times 0.89683 = 0.17937 \\
k_{0,4} &= h_0 f(t_0 + h_0, w_0 + k_{0,3}) = 0.5 f(0.2, 0.17937) \\
&= 0.2 \times 0.78656 = 0.15731 \\
k_0 &= \tfrac{1}{6}(k_{0,1} + 2k_{0,2} + 2k_{0,3} + k_{0,4}) \\
&= \tfrac{1}{6}(0.20000 + 2 \times 0.17897 + 2 \times 0.17937 + 0.15731) \\
&\quad = 0.17900 \\
w_1 &= w_0 + k_0 = 0.0000 + 0.17900 = 0.17900
\end{aligned}$$

Fourth-order Runge-Kutta methods

Example Take $h = 0.2$.

i	t_i	w_i	$k_{i,1}$	$k_{i,2}$	$k_{i,3}$	$k_{i,4}$	k_i
0	0.0	0.00000	0.20000	0.17897	0.17937	0.15731	0.17900
1	0.2	0.17900	0.15734	0.13489	0.13602	0.11422	0.13556
2	0.4	0.31456	0.11427	0.09367	0.09519	0.07618	0.09470
3	0.6	0.40925	0.07626	0.05929	0.06079	0.04568	0.06035
4	0.8	0.46960	0.04576	0.03281	0.03407	0.02284	0.03373
5	1.0	0.50333					

Higher-order Runge-Kutta methods

Low order methods For $n = 1, 2, 3, 4$, there are methods of order n requiring n function evaluations per time step.

Higher order methods At least *six* function evaluations are needed for a *fifth* order method.

Even more evaluations are needed for higher orders.

Choice of order Fourth-order methods usually provide a good balance between accuracy and efficiency.

Comparison of Runge-Kutta methods

Comparison $\dot{y}(t) = e^{-t} - y^2, y(0) = 0.$

	h	$y(0.1)$	$y(1.0)$	N_f	e_{rel}
Euler	0.2		0.564559864473071	5	1.2×10^{-1}
Ralston2	0.2		0.500286600094707	10	6.1×10^{-3}
Heun3	0.2		0.503415367048022	15	1.4×10^{-4}
RK4	0.2		0.503328891202093	20	3.6×10^{-5}
Euler	0.1	0.1000000000000000	0.532904863460103	10	5.9×10^{-2}
Ralston2	0.1	0.0948296905440380	0.502658823715687	20	1.3×10^{-3}
Heun3	0.1	0.0948519042605422	0.503354541136427	30	1.5×10^{-5}
RK4	0.1	0.0948541510517630	0.503345613873078	40	2.8×10^{-6}
Euler	0.025	0.0961469752655123	0.510557320425266	40	1.4×10^{-2}
Ralston2	0.05	0.0948491396932605	0.503183407918572	40	1.6×10^{-4}
Heun3	0.077		0.503350170836445	39	6.3×10^{-6}
RK4	0.1	0.0948541510517630	0.503345613873078	40	1.0×10^{-6}
Exact		$0.094854320\cdots$	$0.5033467\cdots$		

Multistep Methods

Multistep methods

Idea Use information from previous (t_j, w_j).

Stages An n-stage multistep method computes w_{i+1} using $w_i, \ldots, w_{i-n+1}$.

General form The general form of an n-stage *linear* multistep method is

$$w_{i+1} = \sum_{j=0}^{n-1} a_j w_{i-k} + h \sum_{j=0}^{n-1} b_j f(t_{i-j}, w_{i-j}).$$

Adams-Bashforth methods

Derivation

$$y(t_i + h_i) = y(t_i) + \int_{t_i}^{t_i+h_i} \dot{y}(\tau)\, d\tau = y(t_i) + \int_{t_i}^{t_i+h_i} f(\tau, y(\tau))\, d\tau$$

Find a polynomial $p(\tau)$ approximating $f(\tau, y(\tau))$ over $[t_i, t_i + h_i]$.
Estimate $p(\tau)$ by interpolating $f(\tau, y(\tau))$ at $\tau = t_i, t_{i-1}, t_{i-2}, \ldots, t_{i-k+1}$.

Linear interpolation at (t_i, w_i) and (t_{i-1}, w_{i-1}) gives

$$p(\tau) = f(t_i, w_i) + \frac{(\tau - t_i)}{h_{i-1}} \big(f(t_i, w_i) - f(t_{i-1}, w_{i-1}) \big)$$

Then for equally-sized steps,

$$\begin{aligned} y(t_{i+1}) \approx w_{i+1} &= w_i + \int_{t_i}^{t_i+h} f(t_i, w_i) + \frac{(\tau - t_i)}{h} \big(f(t_i, w_i) - f(t_{i-1}, w_{i-1}) \big)\, d\tau \\ &= w_i + h f(t_i, w_i) + \tfrac{1}{2} h \big(f(t_i, w_i) - f(t_{i-1}, w_{i-1}) \big) \\ &= w_i + \tfrac{1}{2} h \big(3 f(t_i, w_i) - f(t_{i-1}, w_{i-1}) \big) \end{aligned}$$

Adams-Bashforth methods

Two-stage Adams-Bashforth method

$$w_{i+1} = w_i + \tfrac{1}{2}h\big(3f(t_i, w_i) - f(t_{i-1}, w_{i-1})\big).$$

Local error $E = \frac{5}{12}h^3 y^{(3)}(\tau)$ for some $\tau \in (t_{i-1}, t_{i+1})$.
Global error $O(h^2)$; second-order.

Three-stage Adams-Bashforth method

$$w_{i+1} = w_i + \tfrac{1}{12}h\big(23f(t_i, w_i) - 16f(t_{i-1}, w_{i-1}) + 5f(t_{i-2}, w_{i-2})\big).$$

Local error $E = \frac{3}{8}h^4 y^{(4)}(\tau)$ for some $\tau \in (t_{i-2}, t_{i+1})$.
Global error $O(h^3)$; third-order.

Four-stage Adams-Bashforth method

$$\begin{aligned} w_{i+1} = w_i + \tfrac{1}{24}h\big(55f(t_i, w_i) &- 59f(t_{i-1}, w_{i-1}) \\ &+ 37f(t_{i-2}, w_{i-2}) - 9f(t_{i-3}, w_{i-3})\big). \end{aligned}$$

Local error $E = \frac{251}{720}h^5 y^{(5)}(\tau)$. Global error $O(h^4)$; fourth-order.

Bootstrapping multistep methods

Bootstrapping The first step(s) of a multistep method require w_i for $i < 0$ which is not available.

e.g.Two-stage Adams-Bashforth:

$$w_1 = w_0 + \tfrac{1}{2}h\big(3f(t_0, w_0) - f(t_{-1}, w_{-1})\big)$$

Use a Runge-Kutta method *of the same order* (or higher) to start the multistep method.

Multistep methods

Example Solve $\dot{y}(t) = e^{-t} - y^2$, $y(0) = 0$ at $t = 1$ using the two-stage Adams-Bashforth method with $h = 0.5$. Booststrap using a second-order method.

Bootstrap using Ralston's second-order method:

$$\begin{aligned} f(t_0, w_0) &= f(0.0, 0.0000) = 1.0000, \\ f\big(t_0 + \tfrac{2}{3}h_0, w_0 + \tfrac{2}{3}h_0 f(t_0, w_0)\big) &= f(0.\dot{3}, 0.3333) = 0.6054, \end{aligned}$$

$$\begin{aligned} y(0.5) \approx w_1 &= w_0 + \tfrac{1}{4}h_0\big(f(t_0, w_0) + 3f(t_0 + \tfrac{2}{3}h_0, w_0 + \tfrac{2}{3}h_0 f(t_0, w_0)\big) \\ &= 0.0000 + \tfrac{1}{4} \times 0.5 \times (1.0000 + 3 \times 0.6054) \\ &= 0.3520 \end{aligned}$$

Take one step of the Adams-Bashforth method to find w_2:

$$\begin{aligned} y(1.0) \approx w_2 &= w_1 + \tfrac{1}{2}h\big(3f(t_1, w_1) - f(t_0, w_0)\big) \\ &= 0.3520 + \tfrac{1}{2} \times 0.5 \times (3 \times f(0.5, 0.3520) - f(0.0, 0.0000)) \\ &= 0.3520 + \tfrac{1}{2} \times 0.5 \times (3 \times 0.48260 - 1.0000) = 0.4640. \end{aligned}$$

Solution $y(1) \approx 0.4640$ has absolute error 0.039, relative error 7.8%.

Multistep methods

Example Solve $\dot{y}(t) = e^{-t} - y^2$, $y(0) = 0$ at $t = 1$ using the two-stage Adams-Bashforth method with $h = 0.1$.

Booststrap using Ralston's second-order method:

$$w_1 = w_0 + \tfrac{1}{4}h_0\big(f(t_0, w_0) + 3f(t_0 + \tfrac{2}{3}h_0, w_0 + \tfrac{2}{3}h_0 f(t_0, w_0)\big).$$

Continue with the two-stage Adams-Bashforth method

$$w_{i+1} = w_i + \tfrac{1}{2}h\big(3f(t_i, w_i) - f(t_{i-1}, w_{i-1})\big).$$

t_i	w_i	$f_i = f(t_i, w_i)$
0.0	0.000000	1.000000
0.1	0.094830	0.895845
0.2	0.179206	0.786616
0.3	0.252407	0.677109
0.4	0.314642	0.571320

t_i	w_i	$f_i = f(t_i, w_i)$
0.5	0.366485	0.472220
0.6	0.408752	0.381734
0.7	0.442401	0.300867
0.8	0.468444	0.229889
0.9	0.487884	0.168539
1.0	0.501670	

So $y(1.0) \approx w_{10} = 0.501670$.

Absolute error 0.0017, relative error 0.33%.

Implicit methods

Implicit methods

Idea Use an extra *forward* interpolation condition $\dot{y}(t_{i+1}) = f(t_{i+1}, w_{i+1})$.

Formula for w_{i+1} then depends on $f(t_{i+1}, w_{i+1})$, leading to an algebraic equation for w_{i+1} which has to be solved.

Adams-Moulton methods

Two-stage interpolation Interpolate $(t_{i-1}, f(t_{i-1}, w_{i-1}))$, $(t_i, f(t_i, w_i))$ and $(t_{i+1}, f(t_{i+1}, w_{i+1}))$. Assume equal step-size h.

$$p(\tau) = f(t_i, w_i) + \frac{(\tau - t_i)}{2h}\big(f(t_{i+1}, w_{i+1}) - f(t_{i-1}, w_{i-1})\big) + \frac{(\tau - t_i)^2}{2h^2}\big(f(t_{i+1}, w_{i+1}) - 2f(t_i, w_i) + f(t_{i-1}, w_{i-1})\big)$$

Then

$$w_i + \int_{t_i}^{t_i+h} p(\tau)\,d\tau = w_i + hf(t_i, w_i) + \tfrac{1}{4}h\big(f(t_{i+1}, w_{i+1}) - f(t_{i-1}, w_{i-1}) + \tfrac{1}{6}h\big(f(t_{i+1}, w_{i+1}) - 2f(t_i, w_i) + f(t_{i-1}, w_{i-1})\big)$$

so

$$w_{i+1} = w_i + \tfrac{1}{12}h\big(5f(t_{i+1}, w_{i+1}) + 8f(t_i, w_i) - f(t_{i-1}, w_{i-1})\big)$$

Adams-Moulton methods

Two-stage implicit Adams-Moulton method

$$w_{i+1} = w_i + \tfrac{1}{12}h\big(5f(t_{i+1}, w_{i+1}) + 8f(t_i, w_i) - f(t_{i-1}, w_{i-1})\big)$$

Local error $E = -\frac{1}{24}h^4 y^{(4)}(\tau)$ for some $\tau \in (t_{i-1}, t_{i+1})$.
Global error $O(h^3)$; third-order.

Three-stage implicit Adams-Moulton method

$$w_{i+1} = w_i + \tfrac{1}{24}h\big(9f(t_{i+1}, w_{i+1}) + 19f(t_i, w_i) - 5f(t_{i-1}, w_{i-1}) + f(t_{i-2}, w_{i-2})\big)$$

Local error $E = -\frac{19}{720}h^5 y^{(5)}(\tau)$ for some $\tau \in (t_{i-2}, t_{i+1})$.
Global error $O(h^4)$; fourth-order.

Explicit versus implicit methods Implicit multistep methods have higher accuracy than explicit methods of the same order. e.g.

Adams-Bashforth three-stage method local error:
$E = \frac{3}{8}y^{(4)}(\tau)h^4$ for some $\tau \in (t_{i-2}, t_{i+1})$.

Adams-Moulton two-stage method local error:
$E = -\frac{1}{24}y^{(4)}(\tau)h^4$ for some $\tau \in (t_{i-1}, t_{i+1})$.

Predictor-corrector approach

Predictor-corrector approach
Use an *estimate* $\tilde{w}_{i+1}$ on the right-hand side of the Adams-Moulton formula.

e.g. $$w_{i+1} = w_i + \tfrac{1}{12}h\big(5f(t_{i+1}, \tilde{w}_{i+1}) + 8f(t_i, w_i) - f(t_{i-1}, w_{i-1})\big)$$

Typically use an explicit Adams-Bashforth method to obtain $\tilde{w}_{i+1}$ with the same number of stages (and lower order).

e.g. $$\tilde{w}_{i+1} = w_i + \tfrac{1}{2}h\big(3f(t_i, w_i) - f(t_{i-1}, w_{i-1})\big).$$

Theorem The combination of an n-stage Adams-Bashforth method (order n) as a predictor with an n-stage Adams-Mouton method (order $n+1$) as a corrector has order $n+1$.

Alternatively, can use an $(n+1)$-stage Adams-Bashforth method and an n-stage Adams-Mouton method to estimate the error.

e.g. $$\tilde{w}_{i+1} = w_i + \tfrac{1}{12}h\big(23f(t_i, w_i) - 16f(t_{i-1}, w_{i-1}) + 5f(t_{i-2}, w_{i-2})\big).$$

May iterate corrector formula to improve accuracy, but improvement is usually not worth the additional effort.

Predictor-corrector approach

Example Solve $\dot{y} = e^{-t} - y^2$, with $y(0) = 0$ using $h = 0.1$.
Assume $y(0.1) \approx 0.09485432$.

Times $t_0 = 0.0,\ t_1 = 0.1,\ t_2 = 0.2, \ldots$ with $w_0 = 0.0$ and $w_1 = 0.09485432$.

Predictor from two-stage Adams-Bashforth method:

$$\tilde{w}_2 = w_1 + \tfrac{1}{2}h\big(3f(t_1, w_1) - f(t_0, w_0)\big) = 0.17923033.$$

Adams-Moulton formula

$$w_{i+1} = w_i + \tfrac{1}{12}h\big(5f(t_{i+1}, \tilde{w}_{i+1}) + 8f(t_i, w_i) - f(t_{i-1}, w_{i-1})\big)$$

Corrector

$$\begin{aligned} w_2 &= w_1 + \tfrac{1}{12}h\big(5f(t_2, \tilde{w}_2) + 8f(t_1, w_1) - f(t_0, w_0)\big) \\ &= 0.0949 + \tfrac{1}{12}\,0.1\big(5f(0.2, 0.1792) + 8f(0.1, 0.0949) - f(0.0, 0.0000)\big) \\ &= 0.0949 + \tfrac{1}{12}\times 0.1\times\big(5\times 0.7866 + 8\times 0.8958 - 1.0000\big) \\ &= 0.17901896. \end{aligned}$$

Applying the corrector again gives 0.17902212.
Repeated application of the corrector converges to 0.17902207.

The exact value is $y(0.2) = 0.17900201$ (8 sf).

Predictor-corrector approach

Example Solve $\dot{y} = e^{-t} - y^2$, with $y(0) = 0$ using $h = 0.1$.
Continue using $t_1 = 0.1,\ w_1 = 0.0949,\ f(t_1, w_1) = 0.8958, t_2 = 0.2,\ w_2 = 0.1790.$

Important: Use the *corrected* value w_2 to compute further in both $\tilde{w}_3$ and w_3!

$$\begin{aligned} \tilde{w}_3 &= w_2 + \tfrac{1}{2}h\big(3f(t_2, w_2) - f(t_1, w_1)\big) \\ &= 0.1790 + \tfrac{1}{2}0.1\big(3f(0.2, 0.1790) - f(0.1, 0.0949)\big) \\ &= 0.1790 + \tfrac{1}{2}\times 0.1\times(3\times 0.7867 - 0.8958) = 0.25222940. \end{aligned}$$

$$\begin{aligned} w_3 &= w_2 + \tfrac{1}{12}h\big(5f(t_3, \tilde{w}_3) + 8f(t_2, w_2) - f(t_1, w_1)\big) \\ &= 0.1790 + \tfrac{1}{12}\,0.1\big(5f(0.3, 0.2522) + 8f(0.2, 0.1790) - f(0.1, 0.0949)\big) \\ &= 0.1790 + \tfrac{1}{12}\times 0.1\times\big(5\times 0.6772 + 8\times 0.7867 - 0.8958\big) = 0.25221576. \\ \tilde{w}_4 &= w_3 + \tfrac{1}{2}h\big(3f(t_3, w_3) - f(t_2, w_2)\big) \\ &= 0.2522 + \tfrac{1}{2}0.1\big(3f(0.3, 0.2522) - f(0.2, 0.1790)\big) = 0.31446243 \end{aligned}$$

$$\begin{aligned} w_4 &= w_3 + \tfrac{1}{12}h\big(5f(t_4, \tilde{w}_4) + 8f(t_3, w_3) - f(t_2, w_2)\big) \\ &= 0.2522 + \tfrac{1}{12}\,0.1\big(5f(0.4, 0.3144) + 8f(0.3, 0.2522) - f(0.2, 0.1790)\big) \\ &= 0.2522 + \tfrac{1}{12}\times 0.1\times\big(5\times 0.5714 + 8\times 0.6772 - 0.7867\big) = 0.31461683. \end{aligned}$$

Predictor-corrector approach

Example Solve $\dot{y}(t) = e^{-t} - y^2$, $y(0) = 0$ at $t = 1$ using the two-stage Adams-Bashforth-Moulton predictor-corrector method with $h = 0.1$.

t_i	$\tilde{w}_i$	$\tilde{f}_i = f(t_i, \tilde{w}_i)$	w_i	$f_i = f(t_i, w_i)$
0.0			0.00000000	1.00000000
0.1			0.09485432	0.89584008
0.2	0.17923033	0.78660724	0.17901896	0.78668296
0.3	0.25222940	0.67719855	0.25221576	0.67720543
0.4	0.31446243	0.57143343	0.31461683	0.57133630
0.5	0.36645700	0.47223993	0.36673920	0.47203302
0.6	0.40897734	0.38154917	0.40934481	0.38124846
0.7	0.44293043	0.30039794	0.44334435	0.30003109
0.8	0.46928659	0.22909906	0.46971515	0.22869665
0.9	0.48901809	0.16743097	0.48943762	0.16702048
1.	0.503055859	0.11481424	0.50345044	

$y(1.0) \approx w_{10} = 0.503450$. Absolute error 0.00010, relative error 0.021%.

Fully-implicit approach (Non-examinable)

Fully-implicit approach
Try to solve the implicit equation for w_{i+1} exactly. Need to solve for $z = w_{i+1}$ in

$$z = w_i + \tfrac{1}{12}h\big(5f(t_{i+1}, z) + 8f(t_i, w_i) - f(t_{i-1}, w_{i-1})\big)$$

or equivalantly

$$g_{i+1}(z) = \tfrac{5}{12}hf(t_{i+1}, z) - z + w_i + \tfrac{1}{12}h\big(8f(t_i, w_i) - f(t_{i-1}, w_{i-1})\big) = 0.$$

Typically use the secant method, starting with w_i and $\tilde{w}_{i+1}$, or Newton's method, which requires computing $f_{,y}$ since $g'(z) = \tfrac{5}{12}hf_{,y}(t_{i+1}, z) - 1$.

Gear's method (Non-examinable)

Gear's method Fit a polynomial p to data (t_j, w_j) for $j = i{-}k{+}1, \ldots, i, i{+}1$.
Solve the equation $\dot{p}(t_{i+1}) = f(t_{i+1}, p(t_{i+1}))$.

Backward-difference method For $k = 2$, equally-sized steps h, have

$$p(\tau) = w_i + \frac{\tau - t_i}{2h}(w_{i+1} - w_{i-1}) + \frac{(\tau - t_i)^2}{2h^2}(w_{i+1} - 2w_i + w_{i-1})$$

so

$$\dot{p}(\tau) = \frac{1}{2h}(w_{i+1} - w_{i-1}) + \frac{\tau - t_i}{h^2}(w_{i+1} - 2w_i + w_{i-1}).$$

Taking $\tau = t_{i+1} = t_i + h$ gives

$$\dot{p}(t_{i+1}) = \frac{1}{2h}(w_{i+1} - w_{i-1}) + \frac{1}{h}(w_{i+1} - 2w_i + w_{i-1}) = \frac{1}{2h}(3w_{i+1} - 4w_i + w_{i-1}).$$

So Gear's method is

$$\frac{1}{2h}(3w_{i+1} - 4w_i + w_{i-1}) = f(t_{i+1}, w_{i+1}).$$

Rearranging gives

$$w_{i+1} = \tfrac{4}{3}w_i - \tfrac{1}{3}w_{i-1} + \tfrac{2}{3}hf(t_{i+1}, w_{i+1}).$$

Gear's methods (Non-examinable)

Two-stage Gear's (backward-difference) method

$$w_{i+1} = \tfrac{4}{3}w_i - \tfrac{1}{3}w_{i-1} + \tfrac{2}{3}hf(t_{i+1}, w_{i+1}).$$

Three-stage Gear's (backward-difference) method

$$w_{i+1} = \tfrac{18}{11}w_i - \tfrac{9}{11}w_{i-1} + \tfrac{2}{11}w_{i-2} + \tfrac{6}{11}hf(t_{i+1}, w_{i+1}).$$

Four-stage Gear's (backward-difference) method

$$w_{i+1} = \tfrac{48}{25}w_i - \tfrac{36}{25}w_{i-1} + \tfrac{16}{25}w_{i-2} - \tfrac{3}{25}w_{i-3} + \tfrac{12}{25}hf(t_{i+1}, w_{i+1}).$$

61 / 89

Gear's method (Non-examinable)

Exercise Take $f(t, y) = e^{-t} - y^2$, $h = 0.1$, $w_0 = 0.0$, $w_1 = 0.09485432$. Find w_2 using the backward-difference method.

Solve for $w_2 = z$ in

$$g_2(z) = \tfrac{4}{3}w_1 - \tfrac{1}{3}w_0 + \tfrac{2}{3}hf(t_2, z) - z = 0$$

62 / 89

Stability

63 / 89

Stability analysis

Example Consider $\dot{y} = \lambda y$ with step size h.

Exact solution has

$$y(t + h) = e^{\lambda h}y(t).$$

For $\lambda < 0$, the exact solution is monotonic decreasing.

Euler's method gives

$$w_{i+1} = w_i + hf(w_i) = w_i + h\,\lambda w_i = (1 + h\lambda)w_i.$$

For $-1 < h\lambda < 0$, Euler's method is monotonic decreasing.
For $-2 < h\lambda < -1$, Euler's method oscillates and decreases.
For $-\infty < h\lambda < -2$, Euler's method oscillates and grows.

Say Euler's method is *monotonically stable* for $-1 < h\lambda < 0$, *stable* for $-2 < h\lambda < 0$ and *unstable* for $h\lambda < -2$.

64 / 89

Stability analysis

Linear stability analysis Consider $\dot{y} = \lambda y$ with step size h; set $\alpha = \lambda h$.

Stability criteria
Method is *unconditionally-stable* if $w_i \to 0$ as $i \to \infty$ for all $h\lambda < 0$.
Method is *limit-stable* if $w_{i+1}/w_i \to 0$ as $h\lambda \to -\infty$.

65 / 89

Stability analysis

Stability analyses Euler's method has

$$w_{i+1} = (1+\alpha)w_i,$$

so is monotone and stable for $-1 \leq \alpha \leq 0$, oscillatory and stable for $-2 < \alpha \leq -1$, unstable for $\alpha < -2$.

Ralston's method is stable for $-2 \leq \alpha < 0$, since

$$w_{i+1} = w_i + \tfrac{1}{4}h(\lambda w_i + 3\lambda(w_i + \tfrac{2}{3}\lambda h w_i)) = w_i + \alpha w_i + \tfrac{1}{2}\alpha^2 w_i = (1+\alpha+\tfrac{1}{2}\alpha^2)w_i.$$

Midpoint and modified Euler methods are also stable for $-2 \leq \alpha < 0$.

Third-order Runge-Kutta methods are stable for $-2.51 \leq \alpha < 0$.
Classical fourth-order Runge-Kutta method is stable for $-2.78 \leq \alpha < 0$.

The two-stage Adams-Bashforth method has

$$w_{i+1} = w_i + \tfrac{1}{2}h(3\lambda w_i - \lambda w_{i-1}) = (1+\tfrac{3}{2}\alpha)w_i - \tfrac{1}{2}\alpha w_{i-1}.$$

Assuming $w_i = \beta^i w_0$ gives $\beta^2 - (1+\tfrac{3}{2}\alpha)\beta + \tfrac{1}{2}\alpha = 0$.
Stable for $-0.8 < \alpha < 0$ when $|\beta| < 1$ for all solutions.

Stability analysis

Backward Euler method $w_{i+1} = w_i + hf(t_{i+1}, w_{i+1})$.
Have $w_{i+1} = w_i + h\lambda w_{i+1} = w_i + \alpha w_{i+1}$, so $(1-\alpha)w_{i+1} = w_i$, giving

$$w_{i+1} = \frac{1}{1-\alpha}\, w_i.$$

Unconditionally stable, nonoscillating.

Trapezoid (Crank-Nicolson) method $w_{i+1} = w_i + \tfrac{1}{2}h\big(f(t_i, w_i) + f(t_{i+1}, w_{i+1})\big)$.
Then $w_{i+1} = w_i + \tfrac{1}{2}h(\lambda w_i + \lambda w_{i+1}) = w_i + \tfrac{1}{2}\alpha w_i + \tfrac{1}{2}\alpha w_{i+1}$, so $(1-\alpha/2)w_{i+1} = (1+\alpha/2)w_i$. Hence

$$w_{i+1} = \frac{1+\alpha/2}{1-\alpha/2}\, w_i.$$

Unconditionally stable, oscillates if $\alpha < -2$, ratio $w_{i+1}/w_i \to 1$ as $i \to \infty$.

Stability analysis

Backward-difference method (Non-examinable)

$$w_{i+1} = \tfrac{4}{3}w_i - \tfrac{1}{3}w_{i-1} + \tfrac{2}{3}hf(t_{i+1}, w_{i+1}).$$

Stability $w_i = \beta^i w_0$ yields

$$\beta = \tfrac{4}{3} - \tfrac{1}{3}\beta^{-1} + \tfrac{2}{3}\alpha\beta$$

Hence $(3+2\alpha)\beta^2 - 4\beta + 1 = 0$, so

$$\beta = \frac{4 \pm \sqrt{16 - 4(3+2\alpha)}}{3+2\alpha} = \frac{2 \pm \sqrt{1-2\alpha}}{3+2\alpha} = \frac{1}{2 \mp \sqrt{1-2\alpha}}$$

with $|\beta| < 1$ for $\alpha < 0$.

Unconditionally stable.

Stability analysis

Implicit Adams-Moulton (Non-examinable)
Suppose $w_i = \beta^i w_0$. $f(t,y) = \lambda y$ with $h\lambda = \alpha$. Then

$$\beta w_i = w_i + \tfrac{1}{12}\big(5\alpha\beta w_i + 8\alpha w_i - \alpha w_i/\beta\big).$$

Simplifying gives

$$(12-5\alpha)\beta^2 - (12+8\alpha)\beta + \alpha = 0.$$

Hence

$$\beta = \frac{(12+8\alpha) \pm \sqrt{(12+8\alpha)^2 - 4\alpha(12-5\alpha)}}{2(12-5\alpha)}.$$

This has a positive stable branch and a negative branch which becomes unstable.

If $w_{i-1} = w_i/\gamma$ with $\gamma > 0$, and $w_{i+1} = \beta w_i$, then

$$\beta = \frac{12 + 8\alpha + \alpha/\gamma}{12 - 5\alpha}.$$

In practise, little danger of instability if w_1 is chosen correctly.

Explicit vs implicit methods

Theorem No explicit Runge-Kutta or multistep method is unconditionally stable.

Theorem (Dahlquist barrier) An implicit multistep method can only be unconditionally stable if its order is at most 2.

Stiff systems

Idea A *stiff system* contains *fast* and *stable* variables, which behave similarly to $\dot{y}_i = \lambda y_i$ for $\lambda \ll 0$.

These variables converge quickly to a *pseudo-equilibrium*, after which the dynamics is determined by the *slow* variables.

Explicit methods are unstable for moderate step sizes, requiring either implicit methods, or small step sizes.

Stiff systems

Example Consider the system

$$\dot{y} = 1 - ty.$$

Eventually, any explicit method becomes unstable for any fixed step size!

Example The Fitzhugh-Nagumo equations are

$$\dot{v} = (v - v^3/3 - w + I_{\text{ext}})\tau;$$
$$\dot{w} = (v + a - bw);$$

with parameters

$$I_{\text{ext}} = 0.5;\ a = 0.7;\ b = 0.8;\ \tau = 12.5;$$

The parameter τ describes the *time-scale separation*; the variable v changes quickly and w comparitively slowly.

The Fitzhugh-Nagumo equations are stiff, exhibiting oscillations starting around $h = 0.04$.

72 / 89

Multivariable systems

73 / 89

Multivariable systems

Example Predator-prey system with rabbits (r) and stoats (s) satisfying

$$\dot{r} = r(3 - s); \quad \dot{s} = s(r - 2);$$

with initial conditions

$$r(0) = 5; \quad s(0) = 2.$$

To solve in Matlab, write the system in vector form. Introduce the *state vector* $\mathbf{y}$ with components $y_1 = r$ and $y_2 = s$.

Write in vector form:

$$\dot{\mathbf{y}} = \begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \begin{pmatrix} \dot{r} \\ \dot{s} \end{pmatrix} = \begin{pmatrix} r(3-s) \\ s(r-2) \end{pmatrix} = \begin{pmatrix} y_1(3-y_2) \\ y_2(y_1-2) \end{pmatrix} =: f(t, \mathbf{y}).$$

The initial conditions are

$$\mathbf{y}(0) = \begin{pmatrix} y_1(0) \\ y_2(0) \end{pmatrix} = \begin{pmatrix} r(0) \\ s(0) \end{pmatrix} = \begin{pmatrix} 5 \\ 2 \end{pmatrix} =: \mathbf{y}_0.$$

74 / 89

Multivariable systems

Example Predator-prey system with rabbits (r) and stoats (s) satisfying

$$\dot{r} = r(3-s); \quad \dot{s} = s(r-2); \qquad r(0)=5,\ s(0)=2.$$

Vector form

$$\mathbf{y} = \begin{pmatrix} r \\ s \end{pmatrix}; \quad \dot{\mathbf{y}} = \begin{pmatrix} r(3-s) \\ s(r-2) \end{pmatrix} = \begin{pmatrix} y_1(3-y_2) \\ y_2(y_1-2) \end{pmatrix} =: f(t,\mathbf{y}); \quad \mathbf{y}_0 = \begin{pmatrix} 5 \\ 2 \end{pmatrix}.$$

In Matlab,

```
f=@(t,y)[y(1)*(3-y(2));y(2)*(y(1)-2)]
t0=0, tf=?
y0=[5;2]
```

Solve up to time `tf`:

```
ode45(f,[t0,tf],y0)
```

Alternatively, set up r and s rate equations separately and combine:

```
dotr=@(r,s)r*(3-s), dots=@(r,s)s*(r-2), r0=5, s0=2
f=@(t,y)[dotr(y(1),y(2));dots(y(1),y(2))], y0=[r0;s0]
```


Second-order systems

Example Damped driven pendulum

$$ml^2\ddot{s} + \delta l\dot{s} + mgl\sin(s) = A\cos(\omega t)$$

New variable $v = \dot{s}$ gives equations

$$\dot{s} = v; \quad \dot{v} = \ddot{s} = -(\delta/ml)v - (g/l)\sin(s) + (A/ml^2)\cos(\omega t).$$

Write in vector form $y_1 = s, y_2 = v$:

$$\mathbf{y} = \begin{pmatrix} s \\ v \end{pmatrix}; \quad \dot{\mathbf{y}} = \mathbf{f}(t,\mathbf{y}) = \begin{pmatrix} y_2 \\ (A/ml^2)\cos(\omega t) - (g/l)\sin(y_1) - (\delta/ml)y_2 \end{pmatrix}$$

Run simulation with $m = g = l = 1, \omega = 1,$

$$\dot{y}_1 = y_2; \quad \dot{y}_2 = A\cos(\omega t) - \sin(y_1) - \delta y_2.$$

With $A = 0$, $\delta = 1/4$, $s_0 = 0$, $v_0 = 2$, get damping, slowly varying period.
With $A = 3$, $\delta = 1/4$, $s_0 = 0$ and $v_0 = 2$ get chaos.

Second-order systems

General formulation Suppose

$$\ddot{x} = g(t, x, \dot{x}); \quad x(0) = x_0,\ \dot{x}_0 = v_0.$$

Introduce new variable $v = \dot{x}$ with $\dot{v} = \ddot{x}$ to obtain system

$$\dot{x} = v; \quad \dot{v} = g(t, x, v)$$

Write in vector form with $y_1 = x,\ y_2 = v$ and $\mathbf{y} = \begin{pmatrix} y_1 & y_2 \end{pmatrix}^T$ to obtain

$$\dot{\mathbf{y}} = \begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \begin{pmatrix} \dot{x} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} v \\ g(t, x, v) \end{pmatrix} = \begin{pmatrix} y_2 \\ g(t, y_1, y_2) \end{pmatrix} =: \mathbf{f}(t, \mathbf{y}).$$

Formulation in Matlab

```
g = @(t,x,v) ...
x0=..., v0=...
f = @(t,y) [y(2); g(t,y(1),y(2))]
y0 = [x0;v0]
odeXX(f,[t0,tf],y0)
```


Second-order multivariable systems

Example The motion of a small satellite around a planet of mass M is defined by the differential equation

$$\ddot{\mathbf{x}} = -\frac{GM}{\|\mathbf{x}\|^3}\mathbf{x} =: \mathbf{g}(\mathbf{x})$$

Introduce the velocity $\mathbf{v} = \dot{\mathbf{x}}$. The *state* vector $\mathbf{y}$ is given by $\mathbf{y} = \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix}$.

In components,

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}; \quad \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix}; \qquad \mathbf{y} = \begin{pmatrix} x_1 \\ x_2 \\ v_1 \\ v_2 \end{pmatrix}.$$

The evolution of $\mathbf{y}$ is given by

$$\dot{\mathbf{y}} = \begin{pmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{v}} \end{pmatrix} = \begin{pmatrix} \mathbf{v} \\ \mathbf{g}(\mathbf{x}) \end{pmatrix}$$

Formulation in Matlab

```
g=@(x) (-G*M/norm(x)^3)*x;
f=@(t,y) [y(3:4); g(y(1:2))];
x0=[?;?]; v0=[?;?]; y0=[x0;v0];
```

Adaptive methods

Combine a order-n method $y(t+h) \approx y(t) + k$ with an order-$(n-1)$ method $y(t+h) \approx y(t) + \kappa$.

Expect
$$|y(t+h) - (y+k)| \ll |y(t+h) - (y+\kappa)| \approx |(y+k) - (y+\kappa)| = |k-\kappa|.$$

A *tolerance* of ϵ means that the error of a single step h should be at most ϵh.

A step is *of the embedded lower-order method* acceptable if
$$|k-\kappa| \leq \epsilon h.$$

Matlab Matlab's `ode` methods are all adaptive methods. The `ode23` method is the Bogacki-Shampine method.

Adaptive methods

If $|k-\kappa| > \epsilon h$, the step is rejected and a smaller step size qh is attepted instead.
If $|k-\kappa| \leq \epsilon h$, the step is accepted, and a larger step size of $\max\{h, qh\}$ is used for the *next* step.

For an order-n method with step size qh, if the error for step-size h is approximately $|k-\kappa|$, expect single-step error $|k-\kappa|q^{n+1}$.
Require error for step size qh to be less than $qh\,\epsilon$, so $|k-\kappa|q^{n+1} \leq qh\,\epsilon$.

When taking a step of size h, set
$$q = \left(\frac{\epsilon h}{s|k-\kappa|}\right)^{1/n} \leq \left(\frac{\epsilon h}{|k-\kappa|}\right)^{1/n}$$
Here, s is a safety factor, typically $s = 2$.

Bogacki-Shampine method (Advanced)

Bogacki-Shampine method A third-order method with an embedded second-order method.
$$\begin{aligned}
k_1 &= hf(t, w);\\
k_2 &= hf(t + \tfrac{1}{2}h, w + \tfrac{1}{2}k_1);\\
k_3 &= hf(t + \tfrac{3}{4}h, w + \tfrac{3}{4}k_2);\\
k &= \tfrac{2}{9}k_1 + \tfrac{3}{9}k_2 + \tfrac{4}{9}k_3;\\
k_4 &= hf(t + h, w + \tfrac{2}{9}k_1 + \tfrac{3}{9}k_2 + \tfrac{4}{9}k_3) = hf(t + h, w + k);\\
\kappa &= \tfrac{7}{24}k_1 + \tfrac{6}{24}k_2 + \tfrac{8}{24}k_3 + \tfrac{3}{24}k_4;
\end{aligned}$$
Error estimates
$$y(t+h) - (y(t) + k) = O(h^{3+1}), \qquad y(t+h) - (y(t) + \kappa) = O(h^{2+1}).$$
Note that
$$|\kappa - k| = \left|\tfrac{5}{72}k_1 - \tfrac{1}{12}k_2 - \tfrac{1}{9}k_3 + \tfrac{1}{8}k_4\right| = \tfrac{1}{72}|5k_1 - 6k_3 - 8k_3 + 9k_4|.$$

Bogacki-Shampine method (Advanced)

Example $\dot{y} = e^{-t} - y^2, y(0) = 0, \epsilon = 0.001.$

For $h = 1.0$, find

$$\begin{aligned}
k_{0,1} &= hf(t_0, w_0) = hf(0.00, 0.0000) = 1.0\times 1.0000 = 1.0000;\\
k_{0,2} &= hf(t_0 + \tfrac{1}{2}h, w_0 + \tfrac{1}{2}k_{0,1}) = hf(0.50, 0.5000) = 1.0\times 0.3565 = 0.3565;\\
k_{0,3} &= hf(t_0 + \tfrac{3}{4}h, w_0 + \tfrac{3}{4}k_{0,2}) = hf(0.75, 0.2674) = 1.0\times 0.4009 = 0.4009;\\
k_0 &= \tfrac{2}{9}k_{0,1} + \tfrac{3}{9}k_{0,2} + \tfrac{4}{9}k_{0,3}\\
&= \tfrac{2}{9}\times 1.0000 + \tfrac{3}{9}\times 0.3565 + \tfrac{4}{9}\times 0.4009 = 0.5192;\\
k_{0,4} &= hf(t_0 + h, w_0 + k_0) = hf(1.0, 0.5192) = 1.0\times 0.0983 = 0.0983;\\
\kappa_0 &= \tfrac{7}{24}k_{0,1} + \tfrac{6}{24}k_{0,2} + \tfrac{8}{24}k_{0,3} + \tfrac{3}{24}k_{0,4}\\
&= \tfrac{7}{24}\times 1.0000 + \tfrac{6}{24}\times 0.3565 + \tfrac{8}{24}\times 0.4009 + \tfrac{3}{24}\times 0.0983 = 0.5267;
\end{aligned}$$

Then $|k_0 - \kappa_0| = |0.5192 - 0.5267| = 0.0075$ and $\epsilon h = 0.001$; reject step.

$$q = \left(\tfrac{1}{2}\epsilon h/|k - \kappa|\right)^{1/2} = \left(\tfrac{1}{2}\times 0.001\times 1.0/0.0075\right)^{1/2} = 0.067^{1/2} \approx 0.26.$$

Try step size $\hat{h} = qh = 0.26\times 1.00 = 0.26$.

Bogacki-Shampine method (Advanced)

Example $\dot{y} = e^{-t} - y^2, y(0) = 0, \epsilon = 0.001.$

For $h = 0.26$, find

$$\begin{aligned}
k_{0,1} &= hf(t_0, w_0) = hf(0.00, 0.0000) = 0.26\times 1.0000 = 0.2686;\\
k_{0,2} &= hf(t_0 + \tfrac{1}{2}h, w_0 + \tfrac{1}{2}k_{0,1}) = hf(0.13, 0.1293) = 0.26\times 0.8620 = 0.2229;\\
k_{0,3} &= hf(t_0 + \tfrac{3}{4}h, w_0 + \tfrac{3}{4}k_{0,2}) = hf(0.19, 0.1672) = 0.26\times 0.7958 = 0.2058;\\
k_0 &= \tfrac{2}{9}k_{0,1} + \tfrac{3}{9}k_{0,2} + \tfrac{4}{9}k_{0,3}\\
&= \tfrac{2}{9}\times 0.2686 + \tfrac{3}{9}\times 0.2229 + \tfrac{4}{9}\times 0.2058 = 0.2232;\\
k_{0,4} &= hf(t_0 + h, w_0 + k_0) = hf(0.26, 0.2232) = 0.26\times 0.7223 = 0.1868;\\
\kappa_0 &= \tfrac{7}{24}k_{0,1} + \tfrac{6}{24}k_{0,2} + \tfrac{8}{24}k_{0,3} + \tfrac{3}{24}k_{0,4}\\
&= \tfrac{7}{24}\times 0.2686 + \tfrac{6}{24}\times 0.2229 + \tfrac{8}{24}\times 0.2058 + \tfrac{3}{24}\times 0.1868 = 0.2231;
\end{aligned}$$

Then $|k_0 - \kappa_0| = |0.22321 - 0.22308| = 1.3\times 10^{-4} \leq \epsilon h = 2.6\times 10^{-4}$; accept.

$$q = \left(\tfrac{1}{2}\epsilon h/|k - \kappa|\right)^{1/n} = \left(\tfrac{1}{2}\times 0.001\times 0.26/0.00013\right)^{1/2} = 0.97^{1/2} \approx 0.98.$$

Set $t_1 = t_0 + h_0 = 0.26$ and $w_1 = w_0 + k_0 = 0.2232$.
Try next step size $qh = 0.98\times 0.26 = 0.25$.

Bogacki-Shampine method (Advanced)

Example $\dot{y} = e^{-t} - y^2, y(0) = 0, \epsilon = 0.001.$

With $t_1 = 0.26$ (2sf), $w_1 = 0.2232$ (4sf) and $h = 0.25$ (2sf), find

$$\begin{aligned}
k_{1,1} &= hf(t_1, w_1) = hf(0.26, 0.2232) = 0.25\times 0.7223 = 0.1837;\\
k_{1,2} &= hf(t_1 + \tfrac{1}{2}h, w_1 + \tfrac{1}{2}k_{1,1}) = hf(0.39, 0.3150) = 0.25\times 0.5807 = 0.1477;\\
k_{1,3} &= hf(t_1 + \tfrac{3}{4}h, w_1 + \tfrac{3}{4}k_{1,2}) = hf(0.45, 0.3340) = 0.25\times 0.5266 = 0.1339;\\
k_1 &= \tfrac{2}{9}k_{0,1} + \tfrac{3}{9}k_{0,2} + \tfrac{4}{9}k_{1,3}\\
&= \tfrac{2}{9}\times 0.1837 + \tfrac{3}{9}\times 0.1477 + \tfrac{4}{9}\times 0.1339 = 0.1495;\\
k_{1,4} &= hf(t_1 + h, w_1 + k_1) = hf(0.52, 0.3727) = 0.25\times 0.4599 = 0.1169;\\
\kappa_1 &= \tfrac{7}{24}k_{1,1} + \tfrac{6}{24}k_{1,2} + \tfrac{8}{24}k_{1,3} + \tfrac{3}{24}k_{1,4}\\
&= \tfrac{7}{24}\times 0.1837 + \tfrac{6}{24}\times 0.1477 + \tfrac{8}{24}\times 0.1339 + \tfrac{3}{24}\times 0.1169 = 0.1497;
\end{aligned}$$

Then $|k_1 - \kappa_1| = |0.14954 - 0.14973| = 0.00019 < 0.00025 = \epsilon h$; accept.

$$q = \big(\epsilon h/|k - \kappa|\big)^{1/n} = \big(0.001\times 0.25/0.00019\big)^{1/2} = 0.67^{1/2} \approx 0.82.$$

Set $t_2 = t_1 + h_1 = 0.52,\ w_2 = w_1 + k_1 = 0.2232 + 0.1495 = 0.3727.$
Next step $h = qh = 0.82\times 0.25 = 0.21.$

Adaptive multistep methods (Non-examinable)

Changing step size in a multistep method
To halve the step size, in the two-stage Adams Bashforth method, use

$$\begin{aligned}
y(t + \tfrac{1}{2}h) &\approx y(t) + \int_t^{t+h/2} f(t, y(t)) + \frac{(\tau - t)}{h}\big(f(t, y(t)) - f(t-h, y(t-h))\big)\,d\tau\\
&= y(t) + \tfrac{1}{2}hf(t, y(t)) + \tfrac{1}{8}h\big(f(t, y(t)) - f(t-h, y(t-h))\big)\\
&= y(t) + \tfrac{1}{8}h\big(5f(t, y(t)) - f(t-h, y(t-h))\big)
\end{aligned}$$

Hence

$$\begin{aligned}
t_{i+1} &= t_i + h_i \text{ where } h_i = \tfrac{1}{2}h_{i-1}\\
w_{i+1} &= w_i + \tfrac{1}{4}h_i\big(5f(t_i, w_i) - f(t_{i-1}, w_{i-1})\big)
\end{aligned}$$

To double the step size, use

$$\begin{aligned}
t_{i+1} &= t_i + h_i \text{ where } h_i = 2h_{i-1}\\
w_{i+1} &= w_i + h_i\big(2f(t_i, w_i) - f(t_{i-1}, w_{i-1})\big)
\end{aligned}$$

Global errors

Growth of global errors If $\dot{y} = \lambda y$, then $y(t) = y(0)e^{\lambda t}$, so an error of size ϵ at time 0 becomes $\epsilon\, e^{\lambda t}$ at time t.

In general

$$\frac{d}{dt}\big(y_1(t) - y_2(t)\big) = f(t, y_1(t)) - f(t, y_2(t)) = f_{,y}(t, \eta)\,\big(y_1(t) - y_2(t)\big)$$

Error bound Combine global growth of error with local error estimate:

Theorem If $f_y(\tau, y(\tau)) \leq L$ and the local error is $C\, f^{(n)}(\tau, y(\tau))\; h^{n+1}$ then at times $t_n \in [0, T]$,

$$\big|w_n - y(t_n)\big| \leq CM\,\frac{\exp(LT) - 1}{L}\,h^n \text{ where } M = \sup_{\tau\in[0,T]}|f^{(n)}(\tau, y(\tau))|.$$

Global errors

Example (Non-examinable) Duffing oscillator

$$\ddot{x} + \delta\dot{x} + \alpha x + \beta x^3 = \gamma\cos(\omega t).$$

Take $\alpha = \beta = \gamma = \omega = 1,\ \delta = 0.1$; $x(0) = 1.5$, $\dot{x}(0) = 0.0$. Uncontrollable growth of local errors. Chaotic motion.