Computer-Assisted Problem-Solving / Numerical Methods

Ordinary Differential Equations

version: January 31st, 2017

Legend: Method, Theory, Example, Advanced, Appendix

Theory

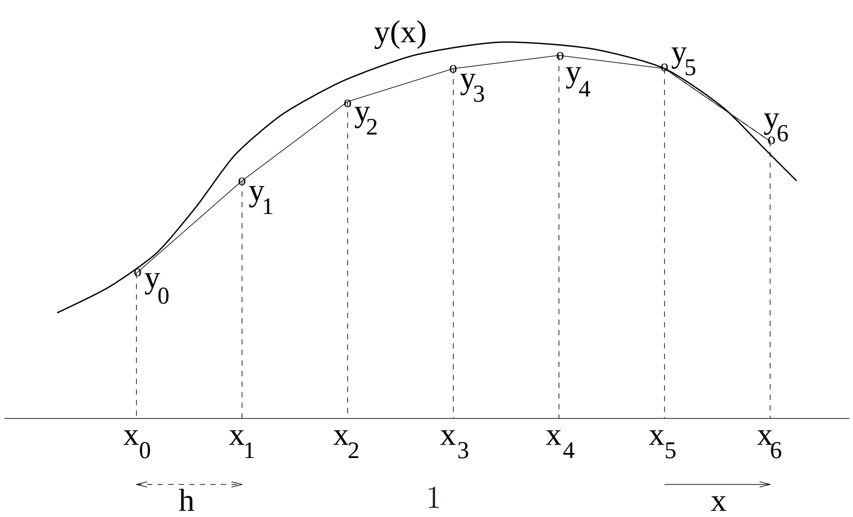
Numerical Methods for ODEs

Often ODEs not analytically solvable,
e.g. $y' = y^2 + x$ with boundary condition $y(0) = 0$

Differential equation: $y'(x) = f(x, y)$
Boundary condition: $y(x_0) = y_0$

Consider finite interval: $x \in [x_0, x_e]$
Divide $[x_0, x_e]$ in N segments, length $h = \dfrac{(x_e - x_0)}{N}$
Grid points: $x_i = x_0 + ih, \quad i = 0, 1, 2, ..., N$

Approximate y in mesh point x_n:
$y_n \approx y(x_n)$, **with y_0 in x_0 known**



Method

Euler's Method

Taylor expansion of $y(x)$ around $x = x_n$:

$$y(x_{n+1}) = y(x_n) + \frac{h}{1!}\frac{dy}{dx}(x_n) + \frac{h^2}{2!}\frac{d^2y}{dx^2}(x_n) + \ldots$$

Neglect 2nd-order terms and higher

Use the ODE: $y'(x_n) = f(x_n, y(x_n))$

Result:

$$y(x_{n+1}) = y(x_n) + hf(x_n, y(x_n)) + R_n,$$

with $R_n = \frac{h^2}{2}y''(\xi)$ **and** $\xi \in [x, x+h]$

Euler's Method:

$$y_{n+1} = y_n + hf(x_n, y_n)$$

Approximation of $y(x)$ in all points:

$$y_0 \rightarrow y_1 \rightarrow y_2 \rightarrow y_3 \rightarrow \ldots \rightarrow y_N$$

Explicit, one-step, 1st-order method

Finite Difference Method:

replace derivative with difference ratio

$$y'(x_n) \approx \frac{(y_{n+1} - y_n)}{h}$$

Example

Example Euler

Solve $y' = y$ **with** $y(0) = 1$

Exact solution: $y(x) = e^x$

Euler's method: $y_{n+1} = y_n + hy_n$ **for** $n = 0, 1, 2, \ldots$

Use step sizes $h = 0.2$**,** $h = 0.1$ **and** 0.05

	x_n	y_n	$\epsilon_n = y_n - y(x_n)$
$h = 0.2$	0.4	1.44000	−0.05182
	0.8	2.07360	−0.15194
	1.2	2.98598	−0.33413
	1.6	4.29982	−0.65321
	2.0	6.19174	−1.19732
$h = 0.1$	0.4	1.46410	−0.02772
	0.8	2.14359	−0.08195
	1.2	3.13843	−0.18169
	1.6	4.59497	−0.35806
	2.0	6.72750	−0.66156
$h = 0.05$	0.4	1.47746	−0.01437
	0.8	2.18287	−0.04267
	1.2	3.22510	−0.09502
	1.6	4.76494	−0.18809
	2.0	7.03999	−0.34907

Error decrease $\sim \mathcal{O}(h)$ **when halving** h

Errors increase for $x_n \to \infty$ **!**

Theory

Truncation Errors

Euler's method $y_{n+1} = y_n + hf(x_n, y_n)$
Follows from Taylor series

$$y(x_{n+1}) = y(x_n) + hf(x_n, y(x_n)) + R_n$$

with $R_n = \frac{h^2}{2} y''(\xi)$ **and** $\xi \in [x, x+h]$

If $|y''(x)| \leq M$ **then** $R_n \leq \frac{h^2}{2} M \Longrightarrow$
Discretisation error: $R_n = \mathcal{O}(h^2)$

Also known as '(local) truncation error': error from truncating Taylor series at location x_n

Accumulation of errors in calculation process

$$y_0 \rightarrow y_1 \rightarrow y_2 \rightarrow y_3 \rightarrow \ldots \rightarrow y_N$$

Total error at point x_N

$$\sum_{n=1}^{N} R_n \sim \frac{1}{h} * R_n =: T_n = \mathcal{O}(h)$$

Also known as 'global error'

Since this error is local (at location x_n**) it is sometimes also known as 'local error'**

Theory

Convergence of Euler's Method

Exact solution in points x_n: $y(x_n)$

Numerical solution: y_n

Error in the n-th Euler step: ϵ_n

A method is called convergent if

numerical solution $\rightarrow$ analytical solution

upon grid refinement $h \rightarrow 0$

Formally:

$$\max_n |\epsilon_n| = \max_n |y(x_n) - y_n| \rightarrow 0 \quad \textbf{for} \quad h \rightarrow 0$$

In Appendix A it is shown that

$$|\epsilon_{n+1}| \leq c\,|\epsilon_0| + \mathcal{O}(h), \ \text{ with } c \text{ constant}$$

Conclusion:

Euler's Method is convergent

The error of the method is $\mathcal{O}(h)$

Initial error $c\,|\epsilon_0|$ always remains

Advanced

Consistency of Euler's Method

A method is called consistent if

the correct ODE is solved for $h \to 0$

Formally: $T_n(h) \to 0$ **for** $h \to 0$

Euler's method:

$$\frac{y_{n+1} - y_n}{h} - f(x_n, y_n) = \frac{R_n}{h} = T_n \Longrightarrow$$

Substitute analytical solution:

$$\frac{y(x_{n+1}) - y(x_n)}{h} - y'|_{x_n} = \frac{R_n}{h} = T_n$$

Since $T_n = \mathcal{O}(h) \to 0$ **for** $h \to 0$**,**

Euler's method $\to$ **the correct ODE**

Conclusion: Euler's method is consistent.

Theory

Stability of Euler

Euler's method

$$y_{n+1} = y_n + hf(x_n, y_n)$$

Computing sequence: $y_0 \to y_1 \to y_2 \to \ldots \to y_N$
How strong is the error accumulation?

A method is called numerically stable if a small disturbance in the input y_0 has 'approximately the same' influence on the numerical solution as on the exact solution

Appendix B contains 3 kinds of stability
Difficult to analyse for general problem
In practice, focus on simple test problem

Absolute Stability: consider "test problem"

$$y' = ay, \quad y(x_0) = y_0$$

with a complex constant

Exact (complex) solution: $y(x) = y_0 e^{ax}$

Remark: a **complex** $\Longrightarrow$ **both oscillating and growing/decreasing solutions**

Absolute Stability Region:

combinations of $h > 0$ and complex a for which a disturbance at step n for test problem does not grow at subsequent steps $m \geq n+1$

Application Euler on test problem $y' = ay$ gives

$$y_{n+1} = y_n + hay_n = (1+ah)y_n$$

Define error at x_n: $\delta_n \quad \Longrightarrow y_n = y_{ex}(x_n) + \delta_n$

Error transferred to next step

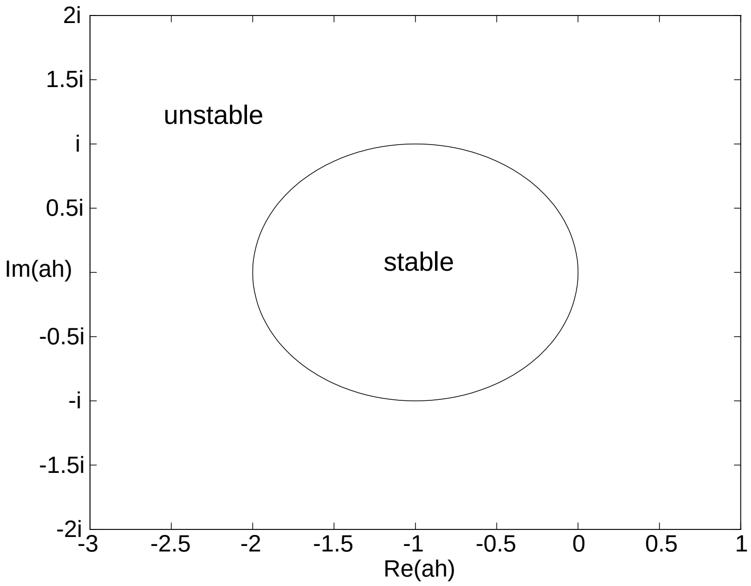
$$\begin{aligned} y_{n+1} &= (1+ah)y_{ex}(x_n) + (1+ah)\delta_n \\ &= y_{ex}(x_{n+1}) + R_n + \delta_{n+1} \end{aligned}$$

New (truncation) error: R_n
Amplification old error: $\delta_{n+1} = (1+ah) * \delta_n$
Amplification factor: $A = 1 + ah$

Absolute stability region of Euler:

$$|1 + ah| \leq 1$$

Inner region and boundary of circle in complex plane with center $(-1, 0)$ and radius 1



Exact solution of test problem: $y(x) = y_0 e^{ax}$

Euler's method gives $y_n = y_0(1 + ah)^n$

1. **For** ah **with** $Re(ah) < 0$ **and outside circle** $|1 + ah| = 1$**:**
   - **Euler gives growing "solution" (in absolute value) for** $x \uparrow$

     since $|1 + ah| > 1 \Rightarrow |1 + ah|^n \uparrow$ **(for** $n \uparrow$**)**
   - **This should be a decreasing solution:** $Re(ah) < 0 \Rightarrow e^{ax} \downarrow$

2. **For** $Re(a) < 0$ **and** h **such that** $ah < -2$
   - **Euler method not absolutely stable**
   - **The term** $(1 + ah)^n$ **changes sign at each step and grows larger (in absolute value)**

     $\Longrightarrow$ **growing oscillating error**

Example

Example Stability of Euler

$y' = -y$, **with** $y(0) = 1$, **exact solution:** $y(x) = e^{-x}$

Euler stable if $|A| = |1 - h| < 1 \Longrightarrow 0 \le h \le 2$

Cases:

$$\begin{array}{llrl} \textbf{(1)} & 0 \le h \le 1 \ : & 0 \le A \le 1, & \textbf{no error amplifier} \\ \textbf{(2)} & 1 < h \le 2 \ : & -1 \le A < 0, & \textbf{no error amplifier,} \\ & & & \textbf{oscillations in solution} \\ \textbf{(3)} & h > 2 \ : & A < -1, & \textbf{error amplification,} \\ & & & \textbf{growing oscill. in sol.} \end{array}$$

Case (1) $0 \le h \le 1$

Results for $h = 0.1$ **en** $h = 0.2$**:**

x_n	$y(x_n)$ **(exact)**	y_n; $h = 0.1$	y_n ; $h = 0.2$
0.0	1.0000000	1	1
0.1	0.9048374	0.9	
0.2	0.8187307	0.81	0.8
0.3	0.7408182	0.729	
0.4	0.6703200	0.6561	0.64
0.5	0.6065306	0.59049	
0.6	0.5488116	0.531441	0.512
0.7	0.4965853	0.4782969	
0.8	0.4493289	0.4304672	0.4096
0.9	0.4065696	0.3874204	
1.0	0.3678794	0.3486784	0.32768

Reasonable results showing good behaviour:
error decreases if $h \to 0$

Case (2) $\quad 1 < h \leq 2$ oscillating solution

x_n	$y(x_n)$ **(exact)**	$y_n;\ h = 1.0$	$y_n;\ h = 1.5$	$y_n;\ h = 2.0$
0.0	1.0000000	1	1	1
1.0	0.3678794	0		
1.5	0.2231301		-0.5	
2.0	0.1353352	0		-1
3.0	0.0497870	0	0.25	
4.0	0.0183156	0		1
4.5	0.0111089		-0.125	
5.0	0.0067379	0		
6.0	0.0024787	0	0.0625	-1

Euler absolutely stable if $0 \leq h \leq 2$**, but .....**

- **If** $h=1$**:** $y_n = y_0(1 + (-1*1))^n = 0$ **for** $n = 1, 2, 3, ..$ **(constant zero solution)**
- **If** $h=2$**:** $y_n = y_0(1 + (-1*2))^n = (-1)^n$ **for** $n = 1, 2, ..$ **(constant oscillation amplitude)**

Solutions are useless for $h = 1$ **or** $h = 2$

Not nice solution, but absolutely stable

Oscillations not always caused by instability

Absolute stability has no implications w.r.t. size of errors or correctness of solution !

Case (3) $h > 2$ growing oscillations

x_n	$y(x_n)$ (exact)	y_n; $h = 2.5$	y_n; $h = 5.0$
0.0	1.0000000	1	1
2.5	0.0820849	-1.5	
5.0	0.0067379	2.25	-4
7.5	0.0005530	-3.375	
10.0	0.0000454	5.0625	16

Useless solution, even unstable

Problems increase with increasing h

Theory

Error Estimation for Euler

Truncation error T is of 1st order: $T \sim \mathcal{O}(h)$

We can write

$$T = a_1 h + a_2 h^2 + a_3 h^3 + \ldots.$$

with unknown constants a_1, a_2,...
So, no precise value for error

Estimate error using halving of mesh:
y_{n+1} **solution for step size h**
$\tilde{y}_{n+1}$ **solution for step size $h * 2$**

$$\begin{aligned} y_{n+1} &= y(x_{n+1}) + \alpha_1 h + \alpha_2 h^2 + \alpha_3 h^3 + \ldots \\ \tilde{y}_{n+1} &= y(x_{n+1}) + \alpha_1 2h + \alpha_2 4h^2 + \alpha_3 8h^3 + \ldots \end{aligned}$$

The dominant term in the error, i.e. $\alpha_1 h$, follows from

$$\alpha_1 h + \mathcal{O}(h^2) = \tilde{y}_{n+1} - y_{n+1}$$

If h small $\Longrightarrow$ neglect 2nd-order term $\Longrightarrow$

Result for error in y_{n+1}: $\epsilon_{n+1} \approx \tilde{y}_{n+1} - y_{n+1}$,
i.e. difference between two grids

Example

Error Estimation for Euler

Consider $y' = y$**, with** $y(0)=1$**. Solution:** $y(x)=e^x$
Euler: $y_{n+1} = y_n + hy_n$ **for** $n = 0, 1, 2, \ldots$

	x_n	y_n	$\epsilon_n = y_n - y(x_n)$
$h = 0.2$	0.4	1.44000	-0.05182
	0.8	2.07360	-0.15194
	1.2	2.98598	-0.33413
	1.6	4.29982	-0.65321
$h = 0.1$	0.4	1.46410	-0.02772 *
	0.8	2.14359	-0.08195
	1.2	3.13843	-0.18169
	1.6	4.59497	-0.35806

Estimate error in $y(0.4)$ **with** $h = 0.1$**:**
$1.44000 - 1.46410 = -0.0241$
approximately equal to actual error -0.02772

With this error estimate we can find reasonable choice for h

- **if** h **too large** $\Longrightarrow$ **inaccurate answer**
- **error should be small enough**
- **if** h **much too small** $\Longrightarrow$
  too much computing work and danger of rounding errors

Approach: refine mesh number of times and study resulting errors

Method

Richardson Extrapolation

Extrapolation of solution for h and $h/2$

Result: higher convergence order obtained

Exact solution $y(x)$

Numerical solution for h and $h/2$: y_h and $y_{h/2}$

When method has $\mathcal{O}(h)$ behaviour $\Longrightarrow$

$$\begin{aligned} y(x) &= y_h(x) + \alpha h + \mathcal{O}(h^2) \\ y(x) &= y_{h/2}(x) + \alpha \frac{h}{2} + \mathcal{O}(h^2) \end{aligned}$$

Combining gives Extrapolation for Euler

$$y(x) = 2y_{h/2}(x) - y_h(x) + \mathcal{O}(h^2)$$

When method has $\mathcal{O}(h^2)$ behaviour $\Longrightarrow$

$$\begin{aligned} y(x) &= y_h(x) + \alpha h^2 + \mathcal{O}(h^3) \\ y(x) &= y_{h/2}(x) + \alpha \frac{h^2}{4} + \mathcal{O}(h^3) \end{aligned}$$

Combining gives Extrapolation formula

$$y(x) = \frac{4}{3} y_{h/2}(x) - \frac{1}{3} y_h(x) + \mathcal{O}(h^3)$$

Error estimate: $\epsilon_{h/2}(x) := y(x) - y_{h/2}(x)$

$$\begin{aligned} &= \tfrac{1}{3}\{y_{h/2}(x) - y_h(x)\} + \mathcal{O}(h^3) \\ &\approx \tfrac{1}{3}\{y_{h/2}(x) - y_h(x)\} \end{aligned}$$

Often a reasonable estimate

Example

Richardson Extrapolation

Consider $y'(x) = -y^2$, $y(0) = 1$ **on** $[0\,,\,5]$

Exact solution $y(x) = 1/(1+x)$

Use some 2nd order method (not Euler)

Step sizes $h = 0.5$ **and** $h/2 = 0.25$

Solution shown for number of x_n **(not all)**

x_n	$y_h(x_n)$	error	$y_{h/2}(x_n)$	error	$\frac{1}{3}(y_h - y_{h/2})$
1.0	0.483144	0.016856	0.496021	0.003979	0.004292
2.0	0.323610	0.009723	0.330991	0.002342	0.002460
3.0	0.243890	0.006110	0.248521	0.001479	0.001543
4.0	0.195838	0.004162	0.198991	0.001009	0.001051
5.0	0.163658	0.003008	0.165937	0.000730	0.000759

Comparing last two columns $\Longrightarrow$
reasonable error estimate with these two meshes

Better error estimate with smaller mesh size:
$\mathcal{O}(h^3)$**-term smaller (better negligible)**

Extrapolation $\hat{y} = \frac{4}{3}y_{h/2} - \frac{1}{3}y_h$ **gives improvement:**

x_n	$y_h(x_n)$	$y_{h/2}(x_n)$	$\hat{y}$	error in $\hat{y}$
1.0	0.483144	0.496021	0.500313	0.313333E-3
2.0	0.323610	0.330991	0.333451	0.118000E-3
3.0	0.243890	0.248521	0.250065	0.064667E-3
4.0	0.195838	0.198991	0.200042	0.042000E-3
5.0	0.163658	0.165937	0.166697	0.030000E-3

Method

Runge-Kutta Methods

Disadvantages Euler:

- absolute stability region is relatively small
- global error is relatively large

Consequence: step size h has to be small for stability and accuracy

Runge-Kutta methods:

- larger stability region
- smaller truncation error
- more computations at each step
- larger step size h allowed

$\Longrightarrow$ more efficient than Euler

Euler uses "only" $f(x_n, y_n)$

Smaller errors by using f-values in points between x_n and x_{n+1}

Widely used are 2nd-order and 4th-order Runge-Kutta methods (RK2 and RK4)

Method

RK2

With Euler the change of function value of $y(x)$, when going from $x = x_n$ to $x = x_{n+1}$, is equal to $hf(x_n, y_n)$

$$y_{n+1} = y_n + hf(x_n, y_n)$$

RK2 uses weighted average of two estimates k_1 and k_2:

$$y_{n+1} = y_n + w_1 k_1 + w_2 k_2,$$

with

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + \alpha h, y_n + \beta k_1) \end{aligned}$$

Four parameters: w_1, w_2, α and β

The choice $w_1 = 1$, $w_2 = 0 \Longrightarrow$ Euler

Choose parameters such that local truncation error is as small as possible

Theorem: each choice with

$$w_1 + w_2 = 1 \qquad \text{and} \qquad \alpha w_2 = \beta w_2 = \frac{1}{2}$$

gives 2nd-order accuracy

Advanced

Proof

Taylor-expansion (2D) of $f(x,y)$ around (x_n, y_n):

$$\begin{aligned} y_{n+1} &= y_n + w_1 h f(x_n, y_n) + w_2 h f\{x_n + \alpha h, y_n + \beta h f(x_n, y_n)\} \\ &= y_n + w_1 h f_n + w_2 h\{f_n + \alpha h \partial_x f_n + \beta h f_n \partial_y f_n + \mathcal{O}(h^2)\} \\ &= y_n + (w_1 + w_2) h f_n + w_2 h\left(\alpha h \partial_x f_n + \beta h f_n \partial_y f_n\right) + \mathcal{O}(h^3) \end{aligned}$$

with $f_n = f(x_n, y_n)$

Compare this expression with Taylor expansion of $y(x_{n+1})$

$$y(x_{n+1}) = y(x_n) + h y'(x_n) + \frac{h^2}{2} y''(x_n) + \mathcal{O}(h^3).$$

Since $y' = f(x,y) \implies$
$y''(x) = (f(x, y(x)))' = \partial_x f + (\partial_y f) y' = \partial_x f + (\partial_y f) f$
(symbol $'$ **means derivative w.r.t.** x**)**

From this it follows:

$$y(x_{n+1}) = y(x_n) + h f_n + \frac{h^2}{2}\{\partial_x f_n + (\partial_y f_n) f_n\} + \mathcal{O}(h^3).$$

Series up to 2nd-order terms equal if

$$w_1 + w_2 = 1 \quad \text{and} \quad \alpha w_2 = \beta w_2 = \frac{1}{2}$$

This proves the theorem

Remark: impossible to derive 4th expression from condition of equal terms with h^3
(check yourself)

Method

Heun's method

Choice $w_1 = w_2 = \frac{1}{2}$ **and** $\alpha = \beta = 1 \Longrightarrow$

$$y_{n+1} = y_n + \frac{h}{2} f(x_n, y_n) + \frac{h}{2} f(x_{n+1}, y^*_{n+1}),$$

with $y^*_{n+1} = y_n + hf(x_n, y_n)$

Remarks:

- Heun is 2nd order, explicit method

- Heun is predictor-corrector method:

first predict value $\rightarrow$ y^*_{n+1}, **then correct to** y_{n+1}

Algorithm Heun's method:

input: x_0, y_0, h, N

output: y_{n+1} **in points** $x_{n+1} = x_0 + (n+1)h$,

with $n = 0, 1, 2, ...N-1$

for $n = 0, 1, 2, ...N-1$

$x_{n+1} = x_n + h$

$k_1 = hf(x_n, y_n)$

$k_2 = hf(x_{n+1}, y_n + k_1)$

$y_{n+1} = y_n + \frac{1}{2}(k_1 + k_2)$

next n

Example

Example Heun (RK2)

Consider $y'(x) = x + y$**, with** $y(0) = 0$

Exact solution: $y(x) = e^x - x - 1$

Compare solutions Euler and Heun

Step size $h = 0.2$

			Euler		Heun	
n	x_n	exact	y_n	error	y_n	error
0	0.0	0.0000	0.000	0.000	0.0000	0.0000
1	0.2	0.0214	0.000	0.021	0.0200	0.0014
2	0.4	0.0918	0.040	0.052	0.0884	0.0034
3	0.6	0.2221	0.128	0.094	0.2158	0.0063
4	0.8	0.4255	0.274	0.152	0.4153	0.0102
5	1.0	0.7183	0.489	0.229	0.7027	0.0156

Conclusion:

Heun has higher accuracy for given h

Theory

Stability of Heun (RK2)

Heun for test problem $y' = ay \Longrightarrow$

$$\begin{aligned} k_1 &= hay_n \\ k_2 &= ha(y_n + k_1) \end{aligned}$$

Heun's method for this problem

$$\begin{aligned} y_{n+1} &= y_n + \frac{1}{2}ahy_n + \frac{1}{2}ha(y_n + hay_n) \\ &= \left\{ 1 + ah + \frac{1}{2}(ah)^2 \right\} y_n \end{aligned}$$

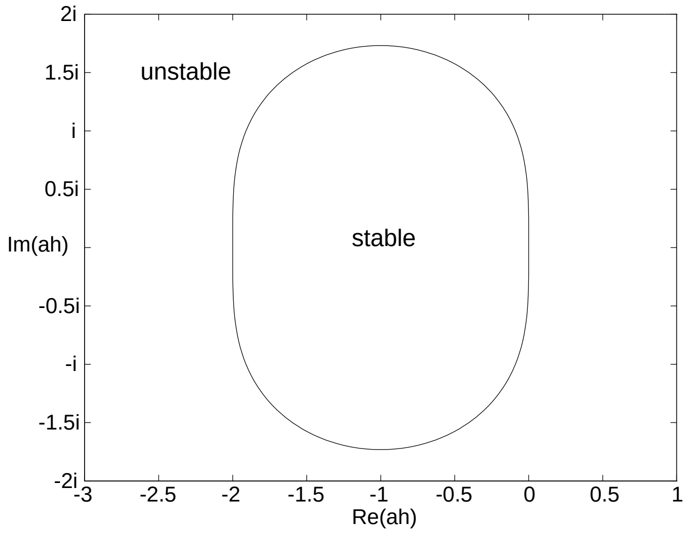
Absolute stability region of Heun:
all ah that satisfy

$$\left| 1 + ah + \frac{1}{2}(ah)^2 \right| \leq 1$$

Remark:
all 2-nd order "two-step" RK-methods have same stability region

Compare this with stability region of Euler:

- **for real a: stability region (on real axis) of Heun is as large as that of Euler**
- **for a with imaginary part $\neq 0$: stability region of Heun larger**



Consequence:
if stability depends on oscillating component in solution $\Longrightarrow$ Heun allows larger steps

Convergence of Heun:
proof analogous to convergence proof Euler

Absolute error $|\epsilon_{n+1}|$ in y_{n+1} for Heun:

$$|\epsilon_{n+1}| \leq c\,|\epsilon_0| + \frac{T^2}{L(1{+}hL/2)}(c-1),$$

with $T^2 = \mathcal{O}(h^2)$ and c and L constant

Thus: global error in y_{n+1} is $\mathcal{O}(h^2)$,

apart from initial error ϵ_0

RK2 is explicit, 2nd-order, ”2-step” method.

Method

RK4

Higher order Runge-Kutta methods use more intermediate points $\in [x_n, x_{n+1}]$

Runge-Kutta method RK4 is very popular

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

with four estimates of $hy'(x)$ for $x \in [x_n, x_{n+1}]$

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \\ k_3 &= hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}) \\ k_4 &= hf(x_{n+1}, y_n + k_3) \end{aligned}$$

RK4 requires per step four computations of function f (to determine $k_1, \dots k_4$) $\Longrightarrow$ compared to Euler 4 times more computations, compared to Heun approximately 2 times more computations per step

Advantages RK4:

- global error in y_{n+1} much smaller

- absolute stability region significantly larger

Example

Example RK4

Consider $y'(x) = x + y$, **with** $y(0) = 0$

Exact solution: $y(x) = e^x - x - 1$

Compare solutions Heun and RK4

Step size $h = 0.2$

			Heun		RK4	
n	x_n	exact	y_n	error	y_n	error
0	0.0	0.000000	0.0000	0.0000	0.000000	0.00E-6
1	0.2	0.021403	0.0200	0.0014	0.021400	3.00E-6
2	0.4	0.091825	0.0884	0.0034	0.091818	7.00E-6
3	0.6	0.222119	0.2158	0.0063	0.222107	1.20E-5
4	0.8	0.425541	0.4153	0.0102	0.425521	2.00E-5
5	1.0	0.718282	0.7027	0.0156	0.718251	3.10E-5

RK4 gives higher accuracy for this (relatively) large h

Theory

Error Estimate RK4

Convergence proof for RK4 $\Longrightarrow$

Global error of RK4 in y_{n+1}

$$|\epsilon_{n+1}| \leq c\,|\epsilon_0| + \frac{T^4}{L(1+\mathcal{O}(hL))}(c-1),$$

with $T^4 = \mathcal{O}(h^4)$ **and** c **and** L **constant**

Error in y_{n+1} **is thus of 4th-order in** h

Consequence:
calculation with step size $2h$ **has roughly 16 times larger error than calculation with** h

Let $\tilde{y}_{n+1}$ **denote the result for** $2h$ **, then**

$$\begin{aligned} y_{n+1} &= y(x_{n+1}) + \alpha_4 h^4 + \alpha_5 h^5 + \alpha_6 h^6 + \ldots \\ \tilde{y}_{n+1} &= y(x_{n+1}) + \alpha_4 16h^4 + \alpha_5 32h^5 + \alpha_6 64h^6 + \ldots \end{aligned}$$

Result: dominant term $\alpha_4 h^4$ **of error becomes**

$$\alpha_4 h^4 + \mathcal{O}(h^5) = \frac{\tilde{y}_{n+1} - y_{n+1}}{15}$$

Often a reasonable estimate of the actual error

Example

Example Error Estimate RK4

$y' = (y - x - 1)^2 + 2$ **with** $y(0) = 1$

RK4 results for $h = 0.1$ **and** $h = 0.2$

x_n	$y(x_n)$ **exact**	y_n $(h{=}0.1)$	$\tilde{y}_n$ $(h{=}0.2)$	$\frac{(\tilde{y}_n - y_n)}{15}$	**absolute error in** y_n
0.0	1.000000000	1.000000000	1.000000000	$0.00E{+}0$	$0.00E{+}0$
0.1	1.200334672	1.200334587			$8.30E{-}8$
0.2	1.402710036	1.402709878	1.402707341	$1.81E{-}7$	$1.57E{-}7$
0.3	1.609336250	1.609336039			$2.10E{-}7$
0.4	1.822793219	1.822792993	1.822788917	$2.91E{-}7$	$2.26E{-}7$

Advanced

Stability of RK4

Apply RK4 for $y' = f(x,y) = ay \Longrightarrow$

$$\begin{aligned} k_1 &= hf(x_n, y_n) &&= hay_n \\ k_2 &= hf(x_n + \tfrac{h}{2}, y_n + \tfrac{k_1}{2}) &&= ha(y_n + \tfrac{k_1}{2}) \\ k_3 &= hf(x_n + \tfrac{h}{2}, y_n + \tfrac{k_2}{2}) &&= ha(y_n + \tfrac{k_2}{2}) \\ k_4 &= hf(x_{n+1}, y_n + k_3) &&= ha(y_n + k_3) \end{aligned}$$

This can be written as

$$\vec{k} = hay_n\vec{e} + ha\,\mathcal{A}\vec{k}$$

$$\vec{k} = \begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{pmatrix} \; ; \; \mathcal{A} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \; ; \; \vec{e} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

From this it follows

$$(I - ha\,\mathcal{A})\vec{k} = hay_n\vec{e}$$

$$\vec{k} = (I - ha\,\mathcal{A})^{-1}hay_n\vec{e}$$

The RK4 formula

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

can be written as

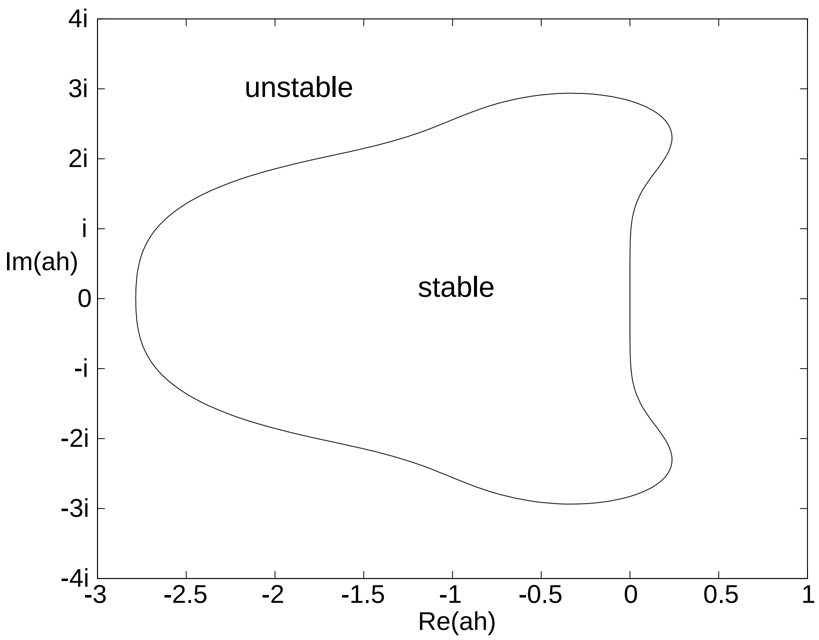
$$y_{n+1} = y_n + \vec{b} \bullet \vec{k} \; ; \; \vec{b} = \begin{pmatrix} \frac{1}{6} \\ \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{6} \end{pmatrix}$$

Combining gives

$$\begin{aligned} y_{n+1} &= y_n + \vec{b} \bullet (I - ha\,\mathcal{A})^{-1} hay_n\vec{e} \\ &= \{\,1 + \vec{b} \bullet (I - ha\,\mathcal{A})^{-1} ha\vec{e}\,\}\, y_n \end{aligned}$$

Amplification factor: $A = |\,1 + \vec{b} \bullet (I - ha\,\mathcal{A})^{-1} ha\vec{e}\,|$

Absolute stability region of RK4:



Advanced

General 4th Order Runge-Kutta

General form

$$y_{n+1} = y_n + w_1 k_1 + w_2 k_2 + w_3 k_3 + w_4 k_4,$$

with

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + \alpha h, y_n + \alpha k_1) \\ k_3 &= hf(x_n + \beta h, y_n + \beta k_2) \\ k_4 &= hf(x_n + \gamma h, y_n + \gamma k_3) \end{aligned}$$

Four parameters: w_1 **t/m** w_4, α, β, γ

By means of Taylor expansions $\Longrightarrow$

4th order accuracy if

$$\begin{aligned} w_1 + w_2 + w_3 + w_4 &= 1 & w_3\alpha\beta + w_4\beta\gamma &= \tfrac{1}{6} \\ w_2\alpha + w_3\beta + w_4\gamma &= \tfrac{1}{2} & w_3\alpha\beta^2 + w_4\beta\gamma^2 &= \tfrac{1}{8} \\ w_2\alpha^2 + w_3\beta^2 + w_4\gamma^2 &= \tfrac{1}{3} & w_3\alpha^2\beta + w_4\beta^2\gamma &= \tfrac{1}{12} \\ w_2\alpha^3 + w_3\beta^3 + w_4\gamma^3 &= \tfrac{1}{4} & w_4\alpha\beta\gamma &= \tfrac{1}{24} \end{aligned}$$

These are 8 eqns. for 7 unknowns

Usual choice

$\alpha = \beta = 1/2$, $\gamma = 1$, $w_1 = w_4 = 1/6$, $w_2 = w_3 = 1/3$

$\Longrightarrow$ **RK4**

Method

Implicit Methods

Explicit method:

computation of new step y_{n+1} only uses evaluations $f(x, y)$ with previously calculated arguments

$\rightarrow$ **e.g. Euler, Heun, Runge-Kutta**

Implicit method:

unknown y_{n+1} itself argument of function f

Taylor expansion of y_n around $x = x_{n+1}$:

$$\begin{aligned} y(x_n) &= y(x_{n+1}) - \frac{h}{1!}\frac{dy}{dx}(x_{n+1}) + \frac{h^2}{2!}\frac{d^2y}{dx^2}(x_{n+1}) - \ldots \\ &+ (-1)^k \frac{h^k}{k!}\frac{d^ky}{dx^k}(x_{n+1}) + \ldots \end{aligned}$$

Neglect 2nd-order and higher terms
$\Longrightarrow$ **Implicit (backward) Euler method:**

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$$

Euler-implicit is 1st- order method, similar to explicit Euler

Local error $\sim \mathcal{O}(h^2) \Longrightarrow$ global error $\sim \mathcal{O}(h)$

Theory

Stability Implicit Euler

Apply Euler-implicit to $y' = ay$

$$y_{n+1} = y_n + hay_{n+1} \Longrightarrow y_{n+1} = \frac{y_n}{(1-ah)}$$

Amplification factor: $A = 1/(1-ah)$

Absolute stability region: all ah **for which**

$$\frac{1}{|1-ah|} \leq 1,$$

i.e. outer region of circle $|1-ah| = 1$

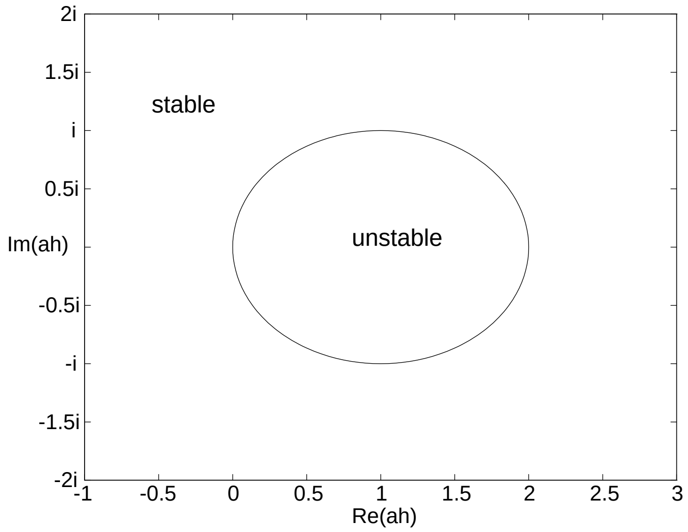
If real part of $ah < 0$**, then real part of** $(1-ah) > 1$ **and thus amplification factor** < 1 **(independent of imaginary part)**

Normally $h > 0$**, so if real part of** $a < 0 \Longrightarrow$ **Euler-implicit absolute stable**

Definition:

Method is A-stable if method is absolute stable for real part $a < 0$

Absolute stability region of Implicit Euler (outer region of the circle)



Advantage:

absolute stability region of Euler-implicit significantly larger than that of Euler-explicit $\Longrightarrow$ larger h possible

Disadvantage:

- **calculation of implicit Euler step requires much more operations**
- y_{n+1} **has to be determined from non-linear relation** $y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$
- **this can be done iteratively (fixed-point method)**

Method

Integration Methods

Implicit Euler: only accuracy $\mathcal{O}(h)$

With integration methods one easily gets higher-order accuracy

Integrating ODE over $[x_n, x_{n+1}]$ gives

$$y(x_{n+1}) - y(x_n) = \int_{x_n}^{x_{n+1}} y' dx = \int_{x_n}^{x_{n+1}} f(x, y) dx$$

Approximate integral with integration method $\Longrightarrow$ (accurate) implicit methods

Example:
Trapezoidal method with locally $\mathcal{O}(h^3)$ gives

$$y_{n+1} - y_n = \frac{h}{2} \left(f(x_n, y_n) + f(x_{n+1}, y_{n+1}) \right)$$

Also known as the Crank-Nicolson method

This method is globally $\mathcal{O}(h^2)$ accurate:

the solution method for $y' = f(x, y)$ "inherits" the order of the integration method

Implicit expression for $y_{n+1} \Longrightarrow$
- fixed point method required
or
- substitute $y_{n+1} \approx y_n + hf(x_n, y_n) \Longrightarrow$ RK2 method

Appendix

A: Convergence of Euler's Method

For convergence we have to show

$$\max_n |\epsilon_n| = \max_n |y(x_n) - y_n| \to 0 \quad \textbf{for} \quad h \to 0$$

Exact solution in points x_n: $y(x_n)$

Numerical solution: y_n

Error in the n-th Euler step: ϵ_n

The Euler method

$$y_{n+1} = y_n + hf(x_n, y_n)$$

follows from

$$y(x_{n+1}) = y(x_n) + hf(x_n, y(x_n)) + hT_n$$

Combination of these gives

$$\epsilon_{n+1} = \epsilon_n + h\{f(x_n, y_n) - f(x_n, y(x_n))\} - hT_n$$

Let $T := \max_n |T_n|$ **(largest truncation error)**

Use triangular inequality $\Longrightarrow$

$$|\epsilon_{n+1}| \leq |\epsilon_n| + h\,|f(x_n, y_n) - f(x_n, y(x_n))| + hT$$

Lemma:
Difference between f-values bounded via Lipschitz-constant $L := \max|\partial f/\partial y|$

$$\begin{aligned} |f(x_n, y_n) - f(x_n, y(x_n))| &= \left| \int_{y(x_n)}^{y_n} \frac{\partial f}{\partial y}(x, \xi) d\xi \right| \\ &\leq L\, |y_n - y(x_n)| \end{aligned}$$

This gives

$$|\epsilon_{n+1}| \leq |\epsilon_n| + hL\, |y_n - y(x_n)| + hT = (1+hL)\, |\epsilon_n| + hT$$

Use this result repeatedly

$$\begin{aligned} |\epsilon_{n+1}| &\leq (1+hL)\, |\varepsilon_n| + hT \\ &\leq (1+hL)\{(1+hL)\, |\epsilon_{n-1}| + hT\} + hT \\ &\leq \ldots \\ &\leq (1+hL)^{n+1}\, |\epsilon_0| \\ &\quad + hT\{1 + (1+hL) + (1+hL)^2 + \ldots + (1+hL)^n\} \\ &= (1+hL)^{n+1}\, |\epsilon_0| + \frac{T}{L}\{(1+hL)^{n+1} - 1\} \end{aligned}$$

The final step follows from

$$1 + r + r^2 + \ldots + r^n = \frac{(r^{n+1} - 1)}{(r - 1)}, \quad \text{with} \quad r = 1+hL$$

Remark:
$h \to 0$ **and** $n \to \infty$ **are equivalent**

Lemma:
$(1+hL)^{n+1}$ **bounded for** $n \to \infty$ **and** $h \to 0$

Proof:
Taylor-series of $\ln(1+hL)$ **around** $h=0$ **gives**

$$\begin{aligned}
(1+hL)^{n+1} &= e^{(n+1)\ln(1+hL)} \\
&= e^{(n+1)(hL+\mathcal{O}(Lh)^2)} \\
&\approx e^{(x_{n+1}-x_0)L} \\
&\leq e^{(x_e-x_0)L} =: c,
\end{aligned}$$

with upper bound c **a positive constant**

Application of this Lemma $\Longrightarrow$
(for h **sufficiently small)**

$$|\epsilon_{n+1}| \leq c\,|\epsilon_0| + \frac{T}{L}(c-1),$$

With c **and** L **constant and** $T = \mathcal{O}(h) \Longrightarrow$

$$|\epsilon_{n+1}| \leq c\,|\epsilon_0| + \mathcal{O}(h)$$

Conclusion:

Euler's Method is convergent

The error of the method is $\mathcal{O}(h)$

Initial error $c\,|\epsilon_0|$ **always remains**

Appendix

B: Definitions of Stability

Euler's method: $y_{n+1} = y_n + hf(x_n, y_n)$
Computing sequence: $y_0 \rightarrow y_1 \rightarrow y_2 \rightarrow \ldots \rightarrow y_N$
How strong is the error accumulation?

A method is called numerically stable if a small disturbance in the input y_0 has 'approximately the same' influence on the numerical solution as on the exact solution

Difficult to analyse for general problem
In practice, focus on simple test problem

ODE $\quad y'(x) = f(x, y) \quad (*)$
Boundary condition: $y(x_0) = y_0$

Let $y^{\epsilon}(x)$ be solution of ODE (*) with disturbance at the boundary $y^{\epsilon}(x_0) = y_0 + \epsilon$

Notice: no disturbance in the ODE, hence $\delta(x) \equiv 0$ in $y'(x) = f(x, y) + \delta(x)$

Remark: if $\epsilon \neq 0 : \exists x \;\; y(x) \neq y^{\epsilon}(x)$

Theorem: $|y^\epsilon(x) - y(x)| \leq c\epsilon, \quad$ **on** $[x_0, x_e]$
with "constant" c **depending on maximum value of** $|\partial f / \partial y|$

In words: the difference between $y^\epsilon(x)$ **and** $y(x)$ **grows linearly, with rate** ϵ

Proof:

Let $\eta(x) := y^\epsilon(x) - y(x)$, **if** $\epsilon = 0 \Longrightarrow \eta(x) \equiv 0$
In point $x = x_0$**:** $\eta(x_0) = \epsilon$ **(by definition)**

For $x \neq x_0$, $\eta'(x)$ **can be written as** $\int \frac{\partial f}{\partial y}$**:**

$$\begin{aligned}
\eta'(x) &= \frac{d}{dx}\left(y^\epsilon(x) - y(x)\right) \\
&= f(x, y^\epsilon) - f(x, y) \\
&= \int_y^{y^\epsilon} \frac{\partial f}{\partial y}(x, \xi) d\xi
\end{aligned}$$

Define the constant of Lipschitz

$$L = \max_{[x_0, x_e]} \left| \frac{\partial f}{\partial y} \right|$$

Find upper bound of $|\eta'|$

$$\begin{aligned}
|\eta'| &= \left| \int_y^{y^\epsilon} \frac{\partial f}{\partial y}(x, \xi) d\xi \right| \leq \int_y^{y^\epsilon} \left| \frac{\partial f}{\partial y}(x, \xi) \right| d\xi \\
&\leq L \left| y^\epsilon(x) - y(x) \right| = L \left| \eta \right|
\end{aligned}$$

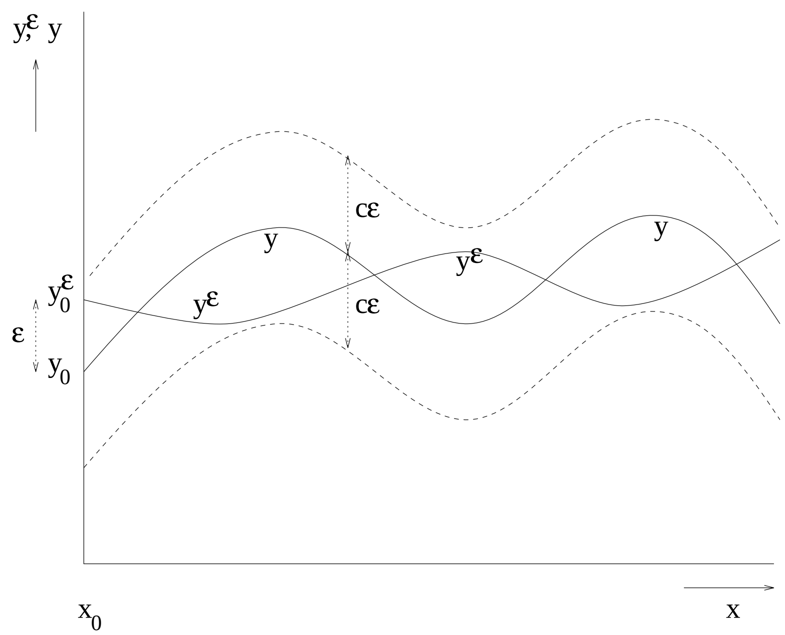
The solution of $\eta' = L\eta$, with $\eta(x_0) = \epsilon$,
is given by $\eta(x) = \epsilon\ e^{L(x-x_0)}$

This leads to
$|\eta| \leq \epsilon\ e^{L(x-x_0)} \leq \epsilon\ \max_x e^{L(x-x_0)} = \epsilon\ e^{L(x_e-x_0)}$

Hence $|y^\epsilon(x) - y(x)| \leq \epsilon\ e^{L(x_e-x_0)} =: \epsilon\ c$

————— end of proof —————

Consequence:
solution of disturbed (analytical) problem y^ϵ
lies within band, with thickness $2c\epsilon$
around exact solution y



Numerical solution in grid points x_n: y_n
Numerical solution disturbed problem: y_n^ϵ

Numerical Stability (definition 1):
(known as zero-stability)

$$|y_n^\epsilon - y_n| \le C\epsilon, \quad \text{for} \quad 1 \le n \le N,$$

with C a constant (independent of ϵ and n)

Problem:
when C very large $\Longrightarrow C\epsilon$ large
$\Longrightarrow$ error in numerical solution large
$\Longrightarrow$ stricter definition needed

Numerical Stability (definition 2):
disturbance δ in computation of y_n
may not lead to an (absolute) disturbance $> \delta$
in y_m, $m > n$ (for any given h)

Now the ODE (function $f(x, y)$) important
$\Longrightarrow$ each problem requires separate check
for numerical stability

Practical approach: Absolute Stability

Numerical Stability (definition 3):
(known as Absolute Stability)

Consider "test problem"

$$y' = ay, \quad y(x_0) = y_0$$

with a complex constant

Exact (complex) solution: $y(x) = y_0 e^{ax}$

Remark: a **complex** $\Longrightarrow$ **both oscillating and growing/decreasing solutions**

Absolute Stability Region:

combinations of $h > 0$ and complex a for which a disturbance at step n for test problem does not grow at subsequent steps $m \geq n+1$

Appendix

C: Simpson Integration

Integration ODE over $[x_n, x_{n+2}]$ **gives**

$$y(x_{n+2}) - y(x_n) = \int_{x_n}^{x_{n+2}} y' dx = \int_{x_n}^{x_{n+2}} f(x,y) dx$$

Use Simpson's integration method, locally $\mathcal{O}(h^3)$

$$\tilde{y}_{n+2} - y_n = \frac{h}{3}\{f(x_n, y_n) + 4f(x_{n+1}, \tilde{y}_{n+1}) + f(x_{n+2}, \tilde{y}_{n+2})\}$$

Unknowns: $\tilde{y}_{n+2}$ **and** $\tilde{y}_{n+1}$

Step 1: Substitute $\tilde{y}_{n+2}$ **in rhs via polynomial** $p_3(x)$ **through** (x_n, y_n) **and** $(x_{n+1}, \tilde{y}_{n+1})$ **with correct tangent (Hermite interpolation)**

$$\begin{aligned} p_3(x) = & \\ & \frac{1}{h^3}\{\, (x - x_{n+1})^2 (2x - 3x_n + x_{n+1}) y_n - (x - x_n)^2 (2x - 3x_{n+1} + x_n) \tilde{y}_{n+1} \,\} \\ + & \frac{1}{h^2}\{\, (x - x_n)(x - x_{n+1})^2 \tilde{y}'(x_n) + (x - x_n)^2 (x - x_{n+1}) \tilde{y}'(x_{n+1}) \,\} \end{aligned}$$

This gives predictor for $\tilde{y}_{n+2}$ **in** $x_{n+2} = x_n + 2h$

$$\begin{aligned} \tilde{y}_{n+2} &\approx p_3(x_{n+2}) \\ &= \ldots = 5y_n - 4\tilde{y}_{n+1} + 2hf(x_n, y_n) + 4hf(x_{n+1}, \tilde{y}_{n+1}), \end{aligned}$$

where $y' = f(x,y)$ **is used in** x_n **and** x_{n+1}

Step 2: Approximate $\tilde{y}_{n+1}$, e.g. with Heun:

$$\begin{aligned} y^{**}_{n+1} &= y_n + hf(x_n, y_n) \\ \tilde{y}_{n+1} &= y_n + \frac{h}{2}\{f(x_n, y_n) + f(x_{n+1}, y^{**}_{n+1})\} \end{aligned}$$

Finally, we obtain

$$y_{n+2} - y_n = \frac{1}{6}(k_1 + 4k_3 + k_4),$$

where

$$\begin{aligned} k_1 &= 2hf(x_n, y_n) \\ k_2 &= 2hf(x_n + h, y_n + \frac{1}{2}k_1) \\ k_3 &= 2hf(x_n + h, y_n + \frac{1}{4}k_1 + \frac{1}{4}k_2) \\ k_4 &= 2hf(x_n + 2h, y_n - k_2 + 2k_3) \end{aligned}$$

This is a Runge-Kutta over 2 steps: $n \to n+2$

Re-formulate via $\hat{h} = 2h$ **into "normal" RK**

Appendix

D: Also Available

Lecture notes Numerical Mathematics (in Dutch):

Multi-step Methods

Adams-Bashforth (AB) methods

Adams-Moulton (AM) methods