Numerická derivace a kvadratura

kvadratura = výpočet určitého integrálu

jednodimenzionální

- Data proložit vhodnou funkcí a tu pak derivovat/integrovat. Vhodné, jsou-li data zatížena chybami. Příklad: Shomateova rovnice

$$C^{\circ}_{pm}(T) = A + BT + CT^2 + DT^3 + E/T^2$$

- Derivace: Nahradit derivace diferencemi
  – potřebujeme několik bodů v okolí
  – přesnost klesá
  Kvadratura: Nahradit součtem přes vybrané body v daném intervalu
  – přesnost stoupe

vícedimenzionální

- parciální derivace: opakuj pro všechny proměnné/směry
- kvadratura: několik vnořených 1D kvadratur (do cca 3D–5D)
  Monte Carlo, Conroy (polopravidelný vzor bodů, něco mezi obdélníkem a MC)

Numerická první derivace

Diferenční vzorce se odvodí z Taylorova rozvoje.

První derivace:

$$\begin{aligned}\frac{f(x+h)-f(x)}{h} &= f'(x) + \frac{h}{2}f''(x) + \frac{h^2}{6}f'''(x) + \cdots \\ &= f'(x) + \mathcal{O}(h) \\ \Rightarrow \quad f'(x) &= \frac{f(x+h)-f(x)}{h} + \mathcal{O}(h)\end{aligned}$$

$g(h) = \mathcal{O}(h^n) \Leftrightarrow$
$\exists M > 0$ a $h_0 > 0$:
$|g(h)| \le Mh^n \ \forall\ h \le h_0$

Je to tedy vzorec 1. řádu v h (chyba je $\mathcal{O}(h)$ neboli řádu h).

Zpřesnění (centrální vzorec)

$$f'(x) = \frac{f(x+h)-f(x-h)}{2h} + \mathcal{O}(h^2)$$

$$f'(x) = \frac{-f(x+2h)+8f(x+h)-8f(x-h)+f(x-2h)}{12h} + \mathcal{O}(h^4) \qquad (1)$$

Derivace zprava – když známe funkci jen na jedné straně — viz mmpc4.mw

$$f'(x) = \frac{-f(x+2h)+4f(x+h)-3f(x)}{2h} + \mathcal{O}(h^2) \qquad (2)$$

Numerická druhá derivace

Stejně odvodíme vzorce pro **druhou derivaci**, nejjednodušší centrální je

$$f''(x) = \frac{f(x+h)-2f(x)+f(x-h)}{h^2} + \mathcal{O}(h^2) \qquad (3)$$

Přesnější centrální vzorec:

$$f''(x) = \frac{-f(x+2h)+16f(x+h)-30f(x)+16f(x-h)-f(x-2h)}{12h^2} + \mathcal{O}(h^4)$$

Jaký krok h zvolit?

Ozn. ε = numerická přesnost: nejmenší zobrazitelné číslo > 1 je $1+\varepsilon$

- 64 bit (`double`, `REAL*8`): $\varepsilon = 2^{-52} \doteq 2\times10^{-16}$, dnešní standard
- 80 bit (extended, `long double`, `REAL*10`): $\varepsilon = 2^{-63} \doteq 1\times10^{-19}$
  (vnitřně používá FPU na x86 architekturách, lze i samostatně)
- 32 bit (`float`, `REAL*4`): $\varepsilon = 2^{-23} \doteq 1\times10^{-7}$
  (dnes minimální zisk rychlosti s výjimkou GPU)

Pravidlo: Nejlepší h má stejnou zaokrouhlovací chybu a chybu metody.

Příklad. Jaké h je optimální v rovnici (1)? — viz mmpc4.mw, Numerical derivative

zaokrouhlovací chyba $\propto \varepsilon/h$
chyba metody $\propto h^4$
$\varepsilon/h = h^4$, pro $\epsilon = 1\times10^{-15} \Rightarrow\ h \approx \varepsilon^{1/5} = \underline{1\times10^{-3}}$

plot/numprec.sh = graf funkce $1+2^x-1$

Numerická kvadratura

Numerická integrace na intervalu $[a,b]$. Předpoklad: dostatečný počet derivací je konečný v celém intervalu. Pokud není (např. $\sqrt{x}$ v intervalu $[0,1]$), nutno řešit substitucí.

Obecný vzorec:

$$\int_a^b f(x)\mathrm{d}x = \sum w_i f(x_i), \quad x_i \in [a,b]$$

Metody:

- ekvidistantní argumenty (Newton–Cotes):
  – uzavřené: používají $f(a)$, $f(b)$
  – otevřené: body jen uvnitř intervalu
- neekvidistantní argumenty (Gauss): obvykle efektivnější (pro složitou funkci a máme-li data v libovolných bodech)

Typicky se interval $[a,b]$ rozdělí na kratší intervaly (případně různě dlouhé) a vhodná metoda se aplikuje v každém z nich.

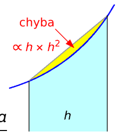
Nevlastní intervaly:

- vhodná substituce převádějící interval na konečný
- speciální metody

Newtonovy–Cotesovy vzorce

Lichoběžníkové pravidlo (uzavřený vzorec):

$$\int_a^b f(x)\mathrm{d}x = \frac{b-a}{2}[f(a)+f(b)] + \mathcal{O}((b-a)^2)$$

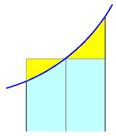


S více dělícími body:

$$\int_a^b f(x)\mathrm{d}x = \left[\frac{f(a)}{2} + f(a+h) + f(a+2h) + \cdots + \frac{f(b)}{2}\right]h + \mathcal{O}(h^2), \quad h = \frac{b-a}{n}$$

Obdélníkové pravidlo (otevřený vzorec): poloviční chyba než lichoběžník!

$$\int_a^b f(x)\mathrm{d}x = (b-a)[f((a+b)/2)] + \mathcal{O}((b-a)^2)$$



Simpsonovo pravidlo:

$$\int_a^b f(x)\mathrm{d}x = \frac{b-a}{6}[f(a)+4f((a+b)/2)+f(b)] + \mathcal{O}((b-a)^4)$$

Výpočet řádu a chyby: z důvodu linearity stačí pro funkce $1, x, x^2, \ldots$

Příklad.

Ověřte, že Simpsonovo pravidlo přesně integruje funkce $1, x, x^2, x^3$, avšak již ne x^4, a proto řád chyby je $\mathcal{O}(h^4)$.

$$\begin{aligned}\int_0^1 1\mathrm{d}x = 1 &= \frac{1-0}{6}(1+4\times1+1) = 1 \\ \int_0^1 x\mathrm{d}x = \frac{1}{2} &= \frac{1-0}{6}(0+4\times\frac{1}{2}+1) = \frac{1}{2} \\ \int_0^1 x^2\mathrm{d}x = \frac{1}{3} &= \frac{1-0}{6}(0+4\times\frac{1}{2^2}+1^2) = \frac{1}{3} \\ \int_0^1 x^3\mathrm{d}x = \frac{1}{4} &= \frac{1-0}{6}(0+4\times\frac{1}{2^3}+1^3) = \frac{1}{4} \\ \int_0^1 x^4\mathrm{d}x = 0.2 &\approx \frac{1-0}{6}(0+4\times\frac{1}{2^4}+1^3) = 0.208\overline{3}\end{aligned}$$

Gaussova(–Legendreova) kvadratura

- Dvoubodový vzorec 4. řádu (2/3 chyba a o bod méně ve srovnání se Simpsonem):
  ověření = domácí úkol

$$\int_a^b f(x)\mathrm{d}x = \frac{b-a}{2}\left[f\left(\frac{a+b}{2} - \frac{a-b}{2\sqrt{3}}\right) + f\left(\frac{a+b}{2} + \frac{a-b}{2\sqrt{3}}\right)\right] + O((b-a)^4) \qquad (4)$$

- Čtyřbodový vzorec 8. řádu (chyba $\mathcal{O}(h^8)$) s dobrou numerickou stabilitou:

```
double Gauss8(double (*f)(double),double a,double b,int n)
// int[a,b] f(x) dx using 4-point Gauss quadrature with n subintervals
{
  const double
    q1=0.43056815579702628761 2, // sqrt((15+sqrt(120))/140)
    q2=0.16999052179242813240 1, // sqrt((15-sqrt(120))/140)
    w=0.17392742256872692868 7; // 1/4-sqrt(5/864)
  double h=(b-a)/n;
  double w1=h*w, w2=h/2-w1;
  double h1=h*q1, h2=h*q2;
  int i;
  double sum=0,x;

  for (i=0; i<n; i++) {
    x=h*(i+0.5)+a;
    sum+=(f(x-h1)+f(x+h1))*w1+(f(x-h2)+f(x+h2))*w2;
  }
  return sum;
}
```

Koeficienty metod jsou kladné, což zajišťuje dobrou numerickou stabilitu.

Richardsonova extrapolace

Symetrické vzorce pro derivace a integrály (a třeba i řešení ODE) mají často chybu tvaru (liché členy jsou nulové)

$$S = S(h) + Ah^n + Bh^{n+2} + \cdots \quad (n \text{ je obv. sudé})$$

Obecně můžeme mít

$$S = S(h) + Ah^n + Bh^{n+1} + \cdots$$

$$\begin{aligned}S &= S(h) + Ah^n \\ S &= S(h/2) + A(h/2)^n\end{aligned}$$

Zpřesnění:

$$S = \frac{\overbrace{2^n S(h/2)}^{S_2(h/2)} - S(h)}{2^n - 1} + \begin{cases}\mathcal{O}(h^{n+2}) \\ \mathcal{O}(h^{n+1})\end{cases}$$

V procesu můžeme pokračovat (s dvojicí $S_2(h/4)$ a $S_2(h/2)$), atd.

- proces selže nebo je nepřesný, nejsou-li splněny předpoklady, totiž že funkce má konečných dost derivací v celém intervalu
- pokud koeficienty u h^k rychle rostou, jsou výsledky více kroků Richardsonovy extrapolace nespolehlivé

Příklady

Příklad. Pomocí vzorce (3) jsme vypočetli druhou derivaci jisté funkce v bodě $x = 0.3$:

h	0.02	0.01
$f''(x)$ podle (3)	74.479013	74.230722

Funkce byla $f(x) = \cos(x)/x$ a $x = 0.3$, pak $f''(0.3) = 74.148327$

Zpřesněte Richardsonovou extrapolací.

Vzorec (3) má chybu $\mathcal{O}(h^2)$. Proto

$$f''(x) = \frac{74.230722\times4 - 74.479013}{4-1} = 74.147958$$

Příklad. Ukažte, že jeden krok Richardsonovy extrapolace lichoběžníkové metody je ekvivalentní Simpsonově metodě.

$$S(1) = \frac{f(0)+f(1)}{2}, \quad S(1/2) = \frac{f(0)+2f(1/2)+f(1)}{4}$$

$$S_{\text{extrap}} = \frac{4S(1/2)-S(1)}{4-1} = \frac{1}{3}\left[\frac{f(0)}{2} + 2f(1/2) + \frac{f(1)}{2}\right]$$

viz mmpc4.mw Numerical quadrature

Obyčejné diferenciální rovnice – počáteční úlohy

$$y' = f(x, y),\ y(x_0) = y_0$$

ODE = Ordinary Differential Equation

- y může být vektor (řešíme soustavu)
- rovnici vyššího řádu můžeme převést na soustavu rovnic 1. řádu (ovšem zpravidla bude efektivnější mít metodu šitou na míru)

$$y'' = f(x, y, y'),\quad \text{subst. } z = y' \ \Rightarrow\ z' = f(x, y, z),\ y' = z$$

- Runge–Kutta: ⊕ nepotřebují historii
  ⊕ snadná změna kroku (i adaptivní)
  ⊕ dobrá stabilita
  ⊖ více výpočtů pravé strany

V tomto kurzu nebude:
- metody pro rovnice vyššího řádu
- okrajové úlohy
- PDE (parciální dif. rov.): metoda sítí, slabá řešení, metoda konečných prvků

- prediktor-korektor: ⊕ efektivnější (méně výpočtů pravé strany)
  ⊖ start – musíme si předem napočítat historii
  ⊖ nesnadná změna kroku
  ⊖ mohou být nestabilní
- metody pro dynamické systémy ($\ddot{r} = f(r, \dot{r}, t)$):
  – symplektické či aspoň reverzibilní (⇒ zachování energie); geometrické integrátory, Trotter
  – metody prediktor–korektor

Rozcvička: Eulerova metoda

Rovnice + počáteční podmínka:

$$y' = f(x, y),\ y(x_0) = y_0$$

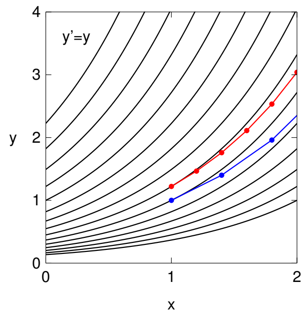
Rozvoj:

$$y(x + h) = y(x) + hy' + \mathcal{O}(h^2)$$

1 krok Eulerovy metody:

$$y(x + h) = y(x) + hf(x, y)$$

Metoda je lokálně 2. řádu, na konečném intervalu musíme provést $\propto 1/h$ kroků, tedy globální chyba je $\mathcal{O}(h)$ (metoda 1. řádu).



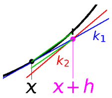
Příklad.
(a) Ukažte, že n kroků Eulerovy metody řešení rovnice $y' = y, y(0) = 1$ s krokem $h = 1/n$ dává $y_n(1) = (1 + 1/n)^n$.
(b) Kolik je $\lim_{n\to\infty} y_n(1)$?
(c) Vypočtěte pro $n = 10$ a $n = 20$ a zpřesněte Richardsonovou extrapolací.

(b) e, (c) 2.712853

Runge–Kutta 2. řádu (RK2)

Zpřesnění (styl lichoběžník) – ve tvaru metody Runge–Kutta (RK2):

$$\begin{aligned} k_1 &:= f(x, y) \\ k_2 &:= f(x + h, y(x) + hk_1) \\ y(x + h) &:= y(x) + \tfrac{h}{2}(k_1 + k_2) \\ x &:= x + h \end{aligned}$$



dohoda: není-li uveden argument funkce, je to x

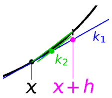
K odvození řádu: $y'' = \mathrm{d}f(x, y)/\mathrm{d}x = f_x + f_y y' = f_x + f_y f \Rightarrow$

$$y(x + h) \;=\; y(x) + \frac{h}{2}(k_1 + k_2)$$
$$\overset{\mathcal{O}(h^3)}{\approx}\; y(x) + \frac{h}{2}(y' + y' + hf_x + hf_y y') = y(x) + hy'(x) + \frac{h^2}{2}y''(x)$$

Tedy metoda má lokální chybu $\mathcal{O}(h^3)$ (nebo lepší) a je (alespoň) 2. řádu.

Jiné **zpřesnění** (styl poloviční krok nebo obdélník):

$$\begin{aligned} k_1 &:= f(x, y) \\ k_2 &:= f(x + \tfrac{h}{2}, y(x) + \tfrac{h}{2}k_1) \\ y(x + h) &:= y(x) + hk_2 \\ x &:= x + h \end{aligned}$$



stejný řád chyby, menší koeficient

Runge–Kutta 3. řádu (RK3)

Jedna z možností, zakončená Simpsonem

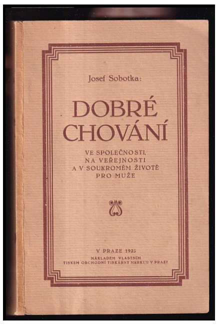
$$\begin{aligned} k_1 &:= f(x, y) \\ k_2 &:= f(x + \tfrac{h}{2}, y(x) + \tfrac{h}{2}k_1) \\ k_3 &:= f(x + h, y(x) + h(2k_2 - k_1)) \\ y(x + h) &:= y(x) + \tfrac{h}{6}(k_1 + 4k_2 + k_3) \\ x &:= x + h \end{aligned}$$

Člen $(2k_2 - k_1)$ je vlastně 1 krok Richardsonovy extrapolace, tj, dává $y'(x + h)$ s větší přesností než oba odhady k_1 a k_2.

Runge–Kutta 4. řádu (RK4)

Populární metoda 4. řádu (lokální chyba $\mathcal{O}(h^5)$) s dobrým chováním:

$$\begin{aligned} k_1 &:= f(x, y) \\ k_2 &:= f(x + \tfrac{h}{2}, y(x) + \tfrac{h}{2}k_1) \\ k_3 &:= f(x + \tfrac{h}{2}, y(x) + \tfrac{h}{2}k_2) \\ k_4 &:= f(x + h, y(x) + hk_3) \\ y(x + h) &:= y(x) + \tfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ x &:= x + h \end{aligned}$$



viz mmpc4.mw Runge-Kutta: $O(h_1)$ – $O(h_4)$

Prediktor-korektor – úvod

Známe historii, tj. hodnoty (a/nebo derivace, tj. pravé strany). S větším množství použité informace lze očekávat přesnější/efektivnější metodu.

- prediktor: predikujeme $y^P(x + h)$:
  – používáme pravou stranu (obv. stabilnější a přesnější)
  – nepoužíváme pravou stranu (Gearovy metody – extrapolujeme funkci i její derivace)
- [případně modifikátor]
- korektor: počítáme výslednou hodnotu $y^C(x + h)$:
  – počítáme pravou stranu jednou
  – počítáme pravou stranu vícktrát
  – počítáme pravou stranu jednou nebo víckrát iteračně

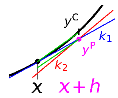
Problém – **stabilita**: chyby v každém kroku se propagují do dalších kroků, metoda musí být navržena tak, že chyby se pokud možno nekumulují a hlavně nerostou exponenciálně.

Pokud koeficienty metody jsou velké a střídají znaménka, metoda bude pravděpodobně nestabilní.

Prediktor-korektor

Nejprve přepišme RK2 do tvaru prediktor-korektor takto:

$$\begin{aligned} y^P(x + h) &= y(x) + hf(x, y(x)) + \mathcal{O}(h^2) \\ y^C(x + h) &= y(x) + \tfrac{h}{2}[f(x, y(x)) + f(x + h, y^P(x + h))] + \mathcal{O}(h^3) \end{aligned}$$



druhý krok je $\mathcal{O}(h^3)$, protože je to lichoběžník, a chyba $y^P(x + h)$ je $h\mathcal{O}(h^2)$.

Zkusme nyní vylepšit oba kroky.

Prediktor:

$$\tfrac{3}{2}y'(x) - \tfrac{1}{2}y'(x) + \tfrac{h}{2}y''(x)$$

$$y^P(x + h) = y(x) + h\left[\tfrac{3}{2}f(x) - \tfrac{1}{2}f(x - h)\right] + \mathcal{O}(h^3)$$

kde $f(x) \equiv f(x, y(x))$ a $f(x - h) \equiv f(x - h, y(x - h))$ (z předchozího kroku).

Korektor hledejme ve tvaru:

$$y^C(x + h) = y(x) + h[af(x - h) + bf(x) + cf(x + h, y^P(x + h))]$$

Metoda testovací rovnice $y = y'$ ($y(x) = \mathrm{e}^x$) ⇒

$$y^C(x + h) = y(x) + \tfrac{h}{12}[-f(x - h) + 8f(x) + 5f(x + h, y^P(x + h)] + \mathcal{O}(h^4)$$

tj. je to metoda 3. řádu (lokální chyba $\mathcal{O}(h^4)$)

viz mmpc4.mw Predictor-corrector – 3rd order example, determining the coefficients of the corrector

Prediktor-korektor 3. řád – stabilita

Metoda je lokálně $\mathcal{O}(h^4)$, takže můžeme napsat (zanedbávajíce $\mathcal{O}(h^5)$):

$$y(x - ih) = y^{\text{přesně}}(x - ih) + \epsilon_i h^4$$

Zkusme testovací rovnici $y' = y$ s $y(0) = 1$ (řešení je $y = \mathrm{e}^x$). Pomocí Maple:

$$\epsilon_{i+1} = \epsilon_i - 13/144$$

- chyba v $y(x - h)$, $y(x - 2h)$... se nešíří dál (s přesností do h^4)
- v jednom kroku vznikne chyba $\propto h^4$

viz mmpc4.mw Predictor–corrector with up to 2 evaluations of the rhs/step: $O(h^1)$ – $O(h^3)$

... a teď se podíváme na to, co se může pokazit

Milneova metoda

Korektor je Simpsonovo pravidlo.

$$\begin{aligned} y^P(x + h) &= y(x - 3h) + \tfrac{4h}{3}[2f(x) - f(x - h) + 2f(x - 2h)] \\ y^C(x + h) &= y(x - h) + \tfrac{h}{3}[f(x - h) + 4f(x) + f(x + h, y^P(x + h)] \end{aligned}$$

Lokální chyba je $\mathcal{O}(h^5)$. Nechť $y(ih)$ má chybu $\epsilon_i h^5$. Pak tato chyba se šíří takto (viz Maple za použití testovací rovnice $y' = y$):

$$\epsilon_{i+1} := \frac{1}{90} + \epsilon_{i-1}$$

Diskuse:

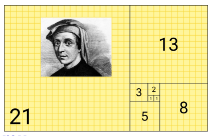
- $\epsilon_{i+1} := \frac{1}{90} + \epsilon_i$ by bylo OK (taková chyba se v principu nedá odstranit)
  příklad nestabilní metody: $\epsilon_{i+1} := \frac{4}{3}\epsilon_i + 1/90$
  příklad stabilní metody: $\epsilon_{i+1} := \frac{3}{4}\epsilon_i + 1/90$
- Růst chyby „ob dva" znamená, že $\epsilon_{\text{sudé}}$ a $\epsilon_{\text{liché}}$ jsou jiné – řešení se může rozkmitat (podle vyšších řádů).
- Milneova metoda je poněkud specifická – na hranici stability

viz mmpc4.mw Milne method and stability

Stabilita a lineární diferenční rovnice

Typicky je rovnice šíření chyb (v řádu lokální chyby) tvaru

$$\epsilon_{i+1} := a_c + a_0\epsilon_i + a_1\epsilon_{i-1} \cdots a_n\epsilon_{i-n}$$



credit: Wikipedia, Oscar en Fotos

To je **lineární diferenční rovnice**. Ta má obecné řešení tvaru

$$\epsilon_i = \sum_x b_x x^i + b_c$$

kde se sčítá se přes všechny kořeny tzv. **charakteristického polynomu**

$$x^{n+1} = c_0 x^n + \cdots + c_n x^0$$

V případě násobných kořenů jsou tam členy x^i, ix^i, atd. – je to podobné, jako pro lineární homogenní diferenciální rovnici.

Chyby nesmí exponenciálně růst ⇒ pro kořeny musí platit $|x| < 1$.

Příklad diferenční rovnice: Fibonacciova posloupnost viz mmpc4.mw

$F_0 = 0,\ F_1 = 1,\ F_n = F_{n-1} + F_{n-2}$ pro $n > 1$

maticově: $\begin{pmatrix} F_{n+2} \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} F_{n+1} \\ F_n \end{pmatrix}$ $\Rightarrow$ $F_n = \dfrac{1}{\sqrt{5}}\left[\left(\dfrac{1+\sqrt{5}}{2}\right)^n - \left(\dfrac{1-\sqrt{5}}{2}\right)^n\right]$