Aproximace funkcí

Numerické metody

6. května 2018

FJFI ČVUT v Praze

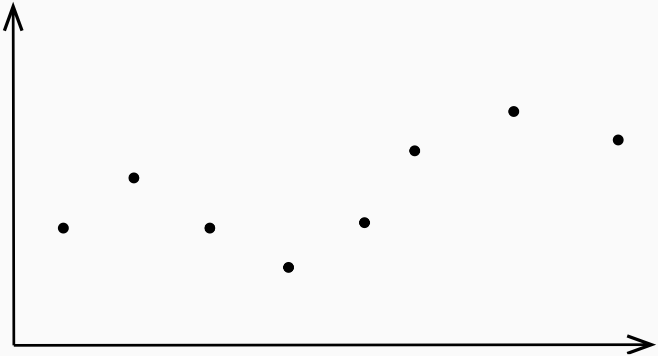
Úvod

Dělení
Interpolace
1D

Více dimenzí
Minimalizace
Důvody

Dělení

zadané data

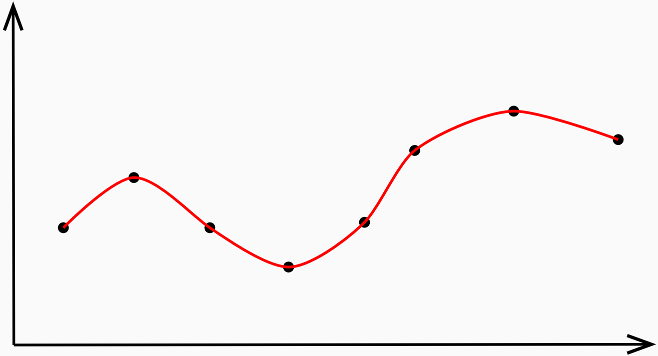


Dělení - Získané data

Obecně máme dva způsoby, jak jsme získali data

1. známe analytický předpis funkce, určíme si diskrétní hodnoty na ose x a v těchto bodech si vypočítáme chování funkce (tedy funkční hodnoty a popřípadně i derivace),
2. máme pouze naměřené hodnoty, nemůžeme si tedy zpětně volit diskrétní hodnoty ani dopočítávat další chování funkce.

interpolace

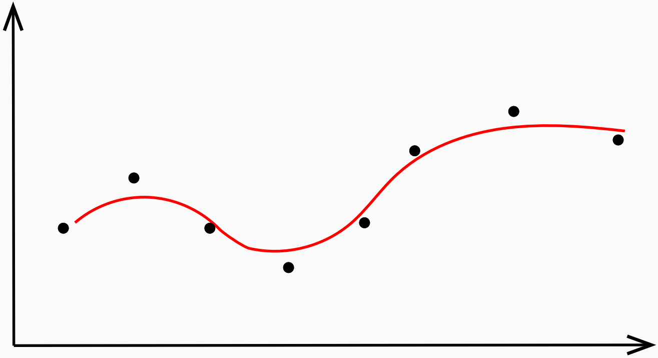


Dělení - Interpolace

Interpolace je taková funkce (křivka), která prochází všemi zadanými body (funkčními hodnotami) a popřípadně v nich má i stejné chování jako původní data (stejné derivace).

Předpokládejme, že máme n bodů, ve každém bodě máme zadánu funkční hodnoty a dalších m podmínek, pak počet volných parametrů (ve funkci, kterou chceme interpolovat) bude často mnohem menší než $n + n \cdot m$ (tedy pro obecný polynom platí, že jeho stupně je mnohem menší než toto číslo).

minimalizace



Dělení - Minimalizace

Oproti interpolaci tato funkce nemusí procházet všemi zadanými body (ani v nich splňovat další podmínky).

Předpokládejme, že mám opět n bodů, ve každém bodě známe funkční hodnoty (tedy máme n hodnot). Pak počet volný parametrů ve funkci bude zpravidla daleko menší než n (tedy kupříkladu stupeň polynomu bude mnohem menší než n).

Interpolace

Interpolace - Úvod

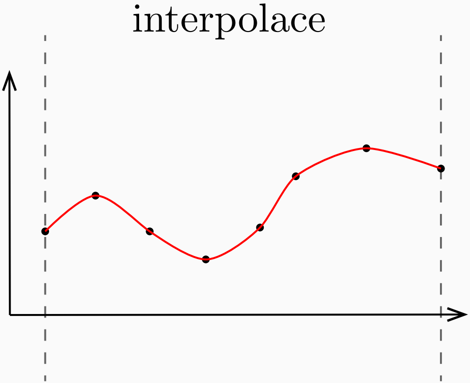
Máme zadané body $x_0 < x_1 < \cdots < x_n$. V těchto bodech známe parametry funkce $f(x)$ (funkční hodnoty f_0, $f_1, \cdots, f_n$, derivace funkce a další).

Hledáme takovou interpolační funkci $\tilde{f}(x)$, která bude mít v daných uzlových bodech stejné chování (paramtery) jak funkce původní.

Mezi těmito body se ale může chovat naprosto jinak.

V numerických metodách tedy hledáme funkční hodnotu $\tilde{f}(x)$ pro libovolné x.

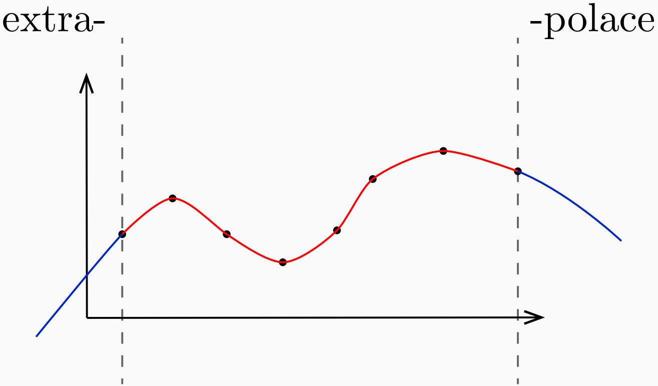
Interpolace - Dělení - extra- vs. inter-



Interpolace - Dělení - interpolace

Pokud hledáme funkční hodnotu $\tilde{f}(x)$ pro $x \in [x_0, x_n]$, tedy uvnitř intervalu, ve kterém interpolujeme, mluvíme o interpolaci.

Interpolace - Dělení - extra- vs. inter-



Interpolace - Dělení - extrapolace

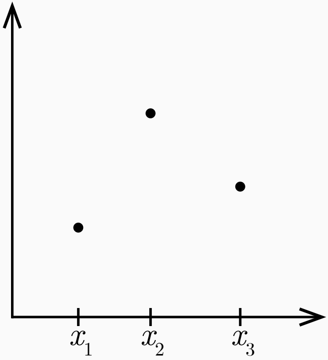
Pokud hledáme funkční hodnotu $\tilde{f}(x)$ pro $x \notin [x_0, x_n]$, tedy mimo interval, ve kterém interpolujeme, mluvíme o extrapolaci.

Extrapolace je často nevyzpytatelná, to znamená, že špatně odhaduje chování funkce.

Interpolace - Dělení - lokálnost

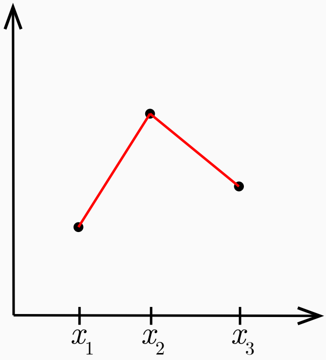
Interpolaci můžeme dále dělit podle toho, zda jednou interpolační funkcí proložíme celý interval $[x_0, x_n]$, pro různé podintervaly použijeme různou interpolační funkci.

zadaná data

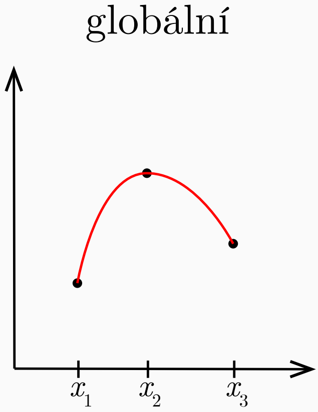


Mějme zadané body a v nich chování funkce.

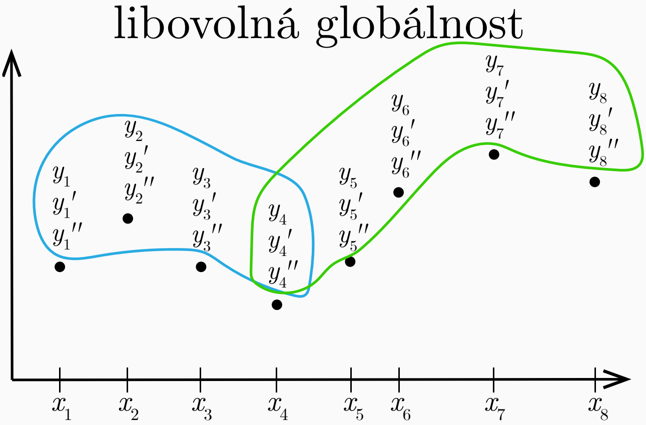
lokální



Interpolaci pak můžeme provést lokálně, tedy kdy bereme v úvahu pouze dva sousední body a ignorovat chování funkce v ostatních bodech.



Nebo naopak můžeme provést globální interpolaci, tedy kdy uvažujeme chování funkce ve všech bodech.



Navíc si můžeme interval rozdělit na podintervaly, ve kterých se už použije globální interpolace (pro daný interval).

1D

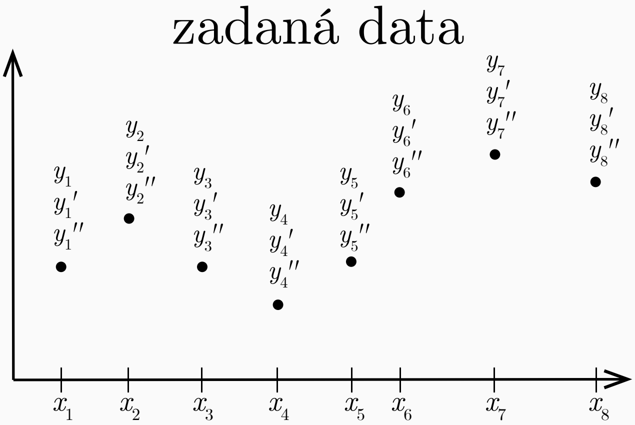
1D - Příklady

Nyní se podíváme na příklady různých interpolacích.

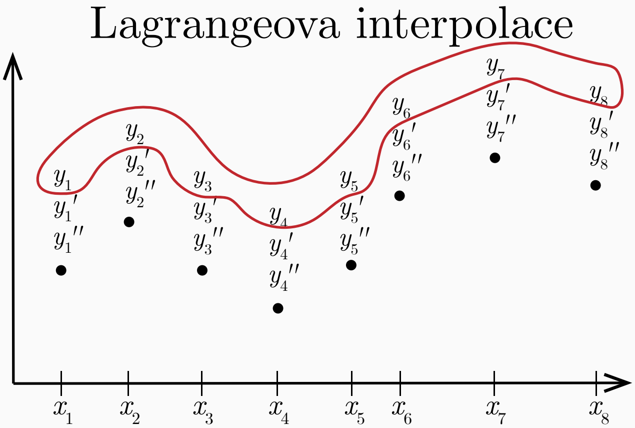
Obecně budeme uvažovat n bodů $x_1 < x_2 < \cdots < x_n$ a v nich zadané parametry funkce (tedy obecně funkční hodnoty a derivace libovolného řádu).

Podle typu interpolace budeme vybírat jak body tak i parametry.

1D - Příklady



Pro příklad různých interpolacích si vezmeme osum bodů a za parametry funkční hodnotu, první a druhou derivaci.



Jednou ze základních interpolacích je Lagrangeova.

1D - Lagrangeova interpolace

Ta záchovává funkční hodnoty. Interpoluje polynomem stupně $n-1$ (tedy v tomto případě polynomem stupně 7).

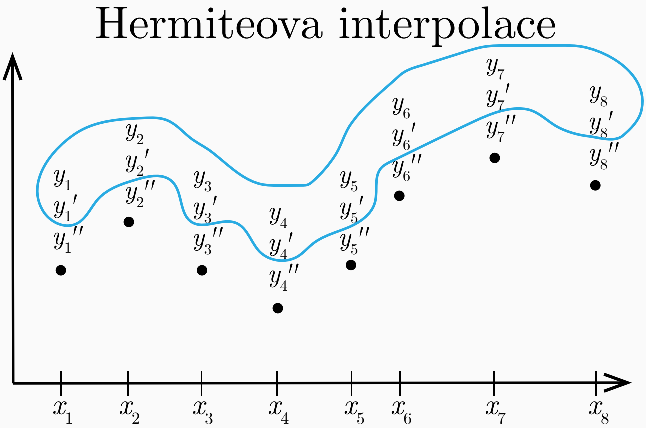
Hledáme tedy takový polynom $\tilde{f}(x) = a_1 + a_2 x + \cdots a_n x^{n-1}$, který splňuje

$$\tilde{f}(x_i) = y_i\,, \qquad \forall i \in \{1, 2, \ldots, n\}\,.$$

1D - Lagrangeova interpolace - konstrukce

Předchozí předpis nám dává možnost konstruovat výsledný polynom pomocí lineárních algebraických rovnic.

To je ale velice zdlouhavé a neefektivní. Nejčastěji se tak Lagrangeova interpolace konstruuje pomocí Nevillova algoritmu.



Jedna z pokročilých interpolací.

1D - Hermiteova interpolace

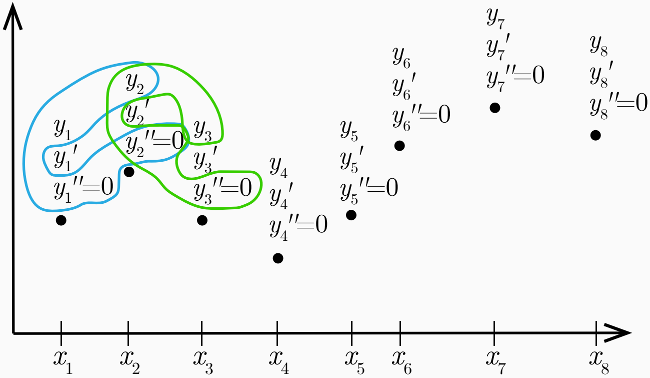
Kromě funkčních hodnot zachovává i derivace. Uvažujme tedy, že máme v každém bodě m parametrů (tedy funkční hodnoty a $m-1$ derivacích nejnižšího řádu), pak stupeň polynomu je $n \cdot m - 1$.

Obecně tedy hledáme polynom $\tilde{f}(x) = a_1 + a_2 x + \cdots a_{n \cdot m} x^{n \cdot m - 1}$, který splňuje

$$\tilde{f}(x_i) = y_i \,,$$
$$\tilde{f}'(x_i) = y_i' \,,$$
$$\vdots$$
$$\tilde{f}^{(m-1)}(x_i) = y_i^{(m-1)}$$

pro $\forall i \in \{1, 2, \ldots, n\}$.

spline



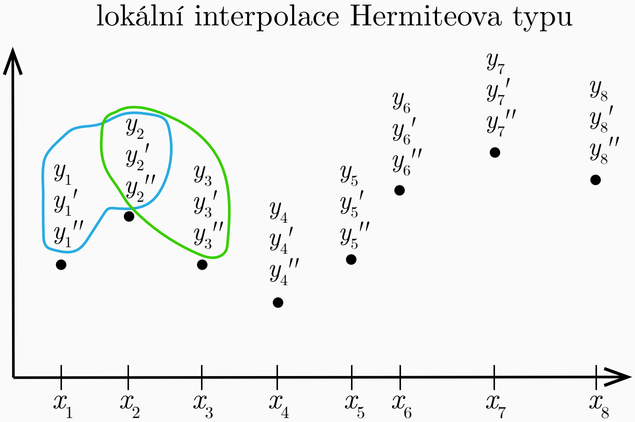
Jedná se o lokální typ interpolace.

1D - Spline

Často se používá, pokud známe jenom funkční hodnoty, ale zároveň chceme, aby (lokální) interpolační funkce byly "hladké" při přechodu mezi intervaly.

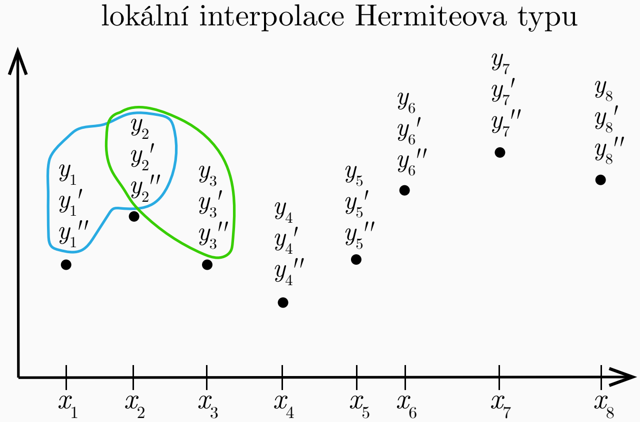
To se nejčastěji dělá tak, že nastavíme druhou derivaci jako nulovou a odvodíme tvar interpolační funkce, která to splňuje.

1D - Lokální interpolace Hermiteova typu



Jedná se o pokročilou lokální interpolaci.

1D - Lokální interpolace Hermiteova typu



Jedná se o pokročilou lokální interpolaci.

1D - Lokální interpolace Hermiteova typu

Tentokrát využíváme znalosti nižších derivací (stejně jako u klasické Hermiteovy interpolace).

Interpolaci ale provádíme lokálně, ale zároveň nám zaručuje spojitost funkčních hodnot a příslušných derivacích při přechodu mezi podintervaly.

1D - Čebyševova interpolace

Předchozí metody mohou používat libovolně rozmístěné body x_i, ale nejčastěji se volí ekvidistantně. To ale vede ke zbytečnému vyčíslování funkce a taky často k oscilacím (pro vyšší stupně polynomu).

Čebyševova interpolace oproti tomu volí body chytře, čímž zamezuje oscilacím a snižuje počet nutných vyhodnocení funkce.

Více dimenzí

Více dimenzí - Úvod

V této úloze se snažím aproximovat funkci více proměnných $f(\vec{x})$.

Konstrukce takové interpolační funkce je náročnější.

Více dimenzí - Možnosti

Jedna z nejjednoušších možností je zafixovat všechny dimenze (tedy zvolit fixní bod) kromě jedné a v ní pak 1D interpolovat. Takto postupovat dokud není provedena interpolace ve všech dimenzích.

Ale často se používají lokální interpolace, nejčastěji spline a lokální interpolace Hermiteova typu.

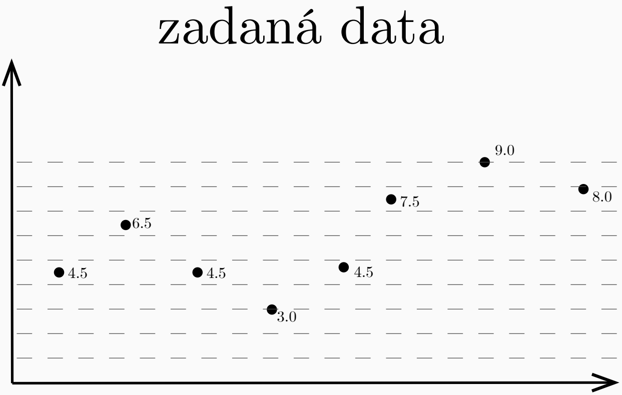
Minimalizace

Minimalizace - Úvod

Oproti interpolaci se nyní snažíme hledat funkci minimalizující chybu v určité třídě funkcí.

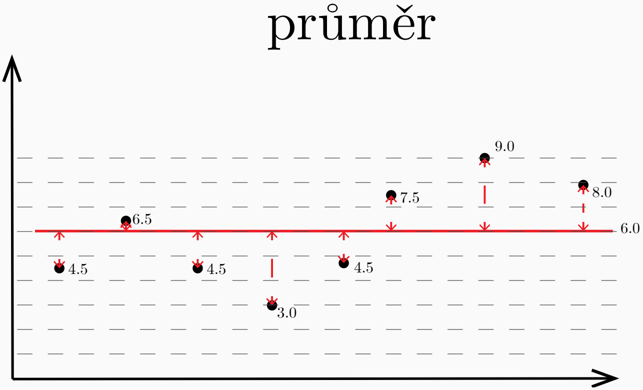
Třída funkcí má povětšinou menší počet volných parametrů, než kolik máme zadaných bodů (oproti interpolace, kde byla rovnost).

Definici chyby si můžeme volit sami, jako příklady mohou složit maximová norma, euklidovská, atd.



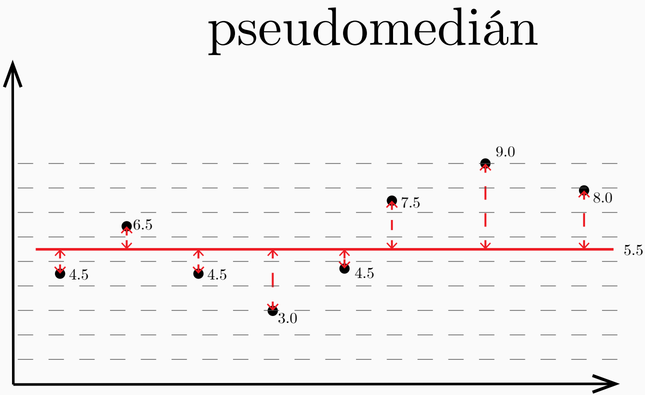
Předpokládejme nyní tyto data, budeme chtít najít konstatní funkci, která minimalizuje chybu.

Minimalizace - Konstantní funkce



Jednou z možností je udělat průměr funkčních hodnot.

Minimalizace - Konstantní funkce



Druhou častou možností je určit medián. Ten je sice méně náchylný na chyby v datech ale je těžší ho spočítat.

Minimalizace - Libovolnou funkcí

Obecně můžeme samožrejmě minimalizovat libovolnou funkci.

Mějme body x_i a původní data y_i, pak se snažíme najít funkci $\phi(x)$, tak aby platilo

$$\phi(x) = \min_{\phi}(\mathbf{error}(y_i, x_i, \phi)),$$

kde $\mathbf{error}$ je obecně definová funkce pro výpočet chyby minimalizace.

Za $\mathbf{error}$ můžeme brát třeba euklidovskou normu

$$\mathbf{error}(y_i, x_i, \phi) = \sqrt{\sum_i (y_i - \phi(x_i))^2}\,.$$

Minimalizace - Libovolnou funkcí

Pro obecnou třídu funkcí (ze které volíme $\phi(x)$) se jedná o relativně složitý problém.

Situace se velice zjednoduší pro třídu polynomů. To vede na soustavu lineárních algebraických rovnic.

Důvody

Důvody - Nevědomost

Máme diskrétní hodnoty (data), neznáme a chceme vědět

- funkční hodnoty mezi diskrétními hodnotami,
- derivaci, ať už v daných bodech nebo mezi body,
- odhad chování funkce (extrapolace pro nekonečno, atd).

Důvody - Náročnost

Funkci známe v libovolném bodě, ale její výpočet je náročný. Chceme tedy funkci aproximovat, tak aby byl výsledek dostatečně dobrý.

Výpočet derivace nebo integrálu (primitivní funkce) je složitý, proto nejdříve funkci aproximujeme něčím, co umíme lehce zintegrovat nebo zderivovat.