

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Číslo projektu	CZ.1.07/1.1.36/02.0066
Autor	Ladislav Kašpárek
Předmět	Programovací metody
Téma	Zobecnění interpolací
Metodický pokyn	výkladový text s ukázkami

Zobecnění interpolací jinými funkcemi

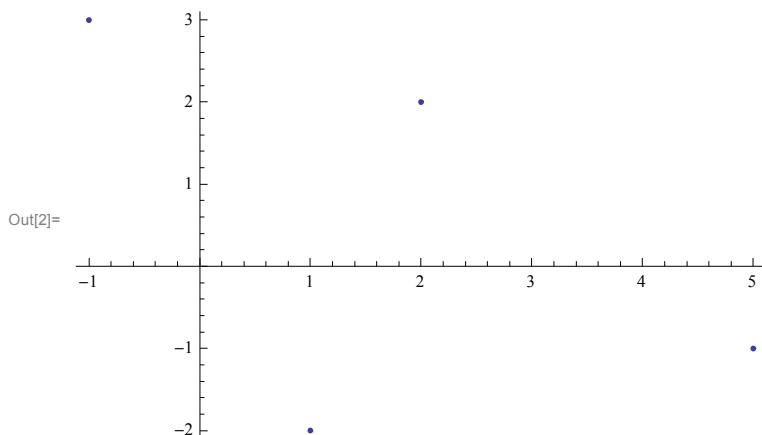
Interpolačních metod je k dispozici celá řada, každá je vhodná pro jiný typ úloh, celá problematika je rozsáhlá a přesahuje naše možnosti. Následující text je rozšířením daného učiva a opírá se o předchozí znalosti řešení soustav rovnic.

Pokračujme ve výkladu o interpolacích pomocí jiných funkcí než polynomu, ale myšlenku poněkud rozšíříme. Dále budeme interpolovat naše 4 známé body:

```
In[1]:= body = {{-1, 3}, {1, -2}, {2, 2}, {5, -1}}
```

```
Out[1]:= {{-1, 3}, {1, -2}, {2, 2}, {5, -1}}
```

```
In[2]:= body // ListPlot
```



Protože již známe mechanismus interpolace polynomem (vede na soustavu lineárních rovnic), snadno zjistíme že Mathematica má funkci, která totéž a se stejným výsledkem dělá také, dále se o její služby také budeme zajímat. Je to funkce: InterpolatingPolynomial.

```
fint[x_] = InterpolatingPolynomial[body, x] // Expand
```

```
Plot[fint[x], {x, -1, 5}]
```

Všechno dohromady (body a graf).

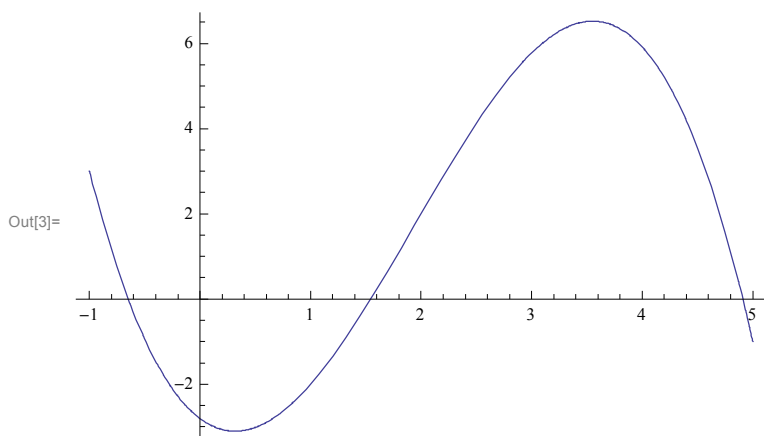
```
Show[Plot[fint[x], {x, -1, 5}],
ListPlot[body, PlotStyle -> Directive[Red, PointSize[Large]]]]
```

Interpolace jinými funkcemi než polynom

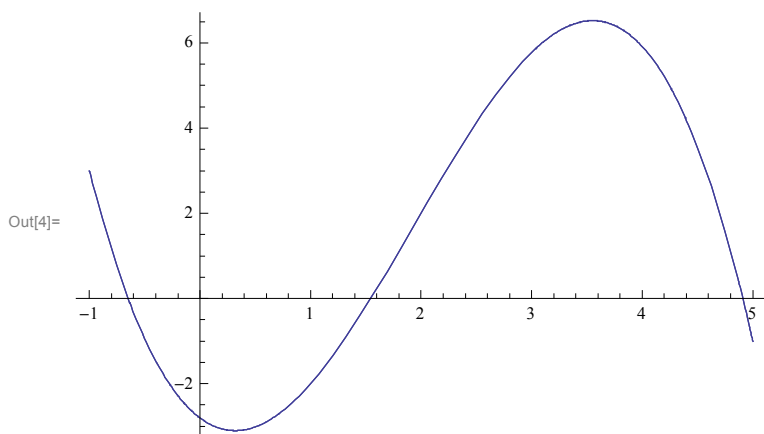
Doposud pro výklad byl použit ruční, mechanický postup výpočtu koeficientů polynomu, aby čtenář pochopil o co v interpolaci polynomem jde a jak proces vypadá.

Celý proces lze zautomatizovat napsáním podprogramu v jazyce systému *Mathematica*, ale to přesahuje rámec výkladu. *Mathematica* má několik vestavěných funkcí pro interpolaci, např: *Interpolation*, která je relativně obecná, neinterpoluje nutně pomocí polynomů (spline, hermite - různé metody) a vrací objekt funkce (její vnitřní postup a výsledek je skrytý). Pro vyšší počet bodů vrací odlišné křivky než newtonova metoda.

```
In[3]:= fint = Interpolation[body, Method -> "Hermite"]; Plot[fint[x], {x, -1, 5}]
```



```
In[4]:= Show[Table[(fint = Interpolation[body, Method -> m]; Plot[fint[x], {x, -1, 5}]),
{m, {"Hermite", "Spline"}}]]
```



```
In[5]:= InterpolatingPolynomial[body, x] // Expand
```

Out[5]=
$$-\frac{101}{36} - \frac{139x}{72} + \frac{119x^2}{36} - \frac{41x^3}{72}$$

Úkol :

Porovnejte graficky výspupy obou funkcí a obou metod interpolace (pro malý počet bodů budou zřejmě totožné). Najděte jinou skupinu bodů, kde budou výstupy různé, experimentujte.

Interpolace goniometrickými funkcemi

Předpokládejme, že funkce kterou hledáme je periodická (z povahy úlohy, kterou řešíme) proto předpis námi hledané funkce může vypadat následovně:

$$f(x) = c \sin(ax + b) + d + c \cos(ax + b)$$

Pro čtyři interpolované body bude tato funkce stačit, pro více bodů bude nutné ji zobecnit, např. pomocí mocnin, apod.

Sestavíme příslušnou soustavu rovnic a pokusíme se najít řešení (algebraické řešení z povahy rovnic nemá smysl hledat, proto numerická metoda – funkce FindRoot). Počáteční hodnoty byly zvoleny náhodně.

```
In[6]:= FindRoot[{3 == c Sin[-1 a + b] + d + c Cos[-1 a + b], -2 == c Sin[1 a + b] + d + c Cos[1 a + b],
  2 == c Sin[2 a + b] + d + c Cos[2 a + b], -1 == c Sin[5 a + b] + d + c Cos[5 a + b]},
  {{a, 1}, {b, 0.5}, {c, -1.1}, {d, 0}}]
```

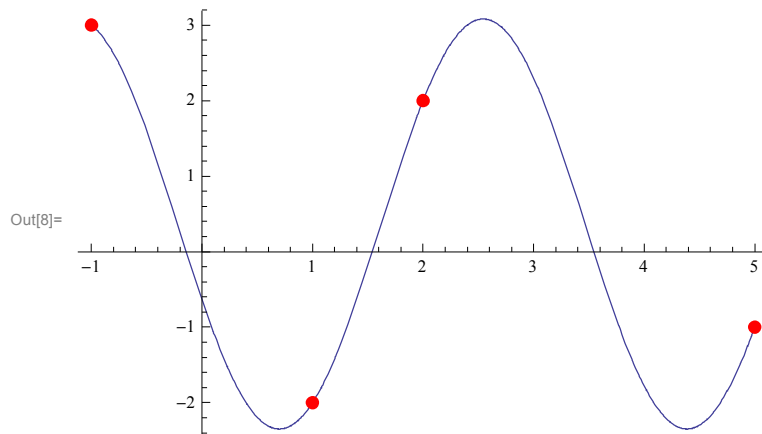
```
Out[6]:= {a -> 1.70351, b -> -0.405582, c -> -1.92031, d -> 0.366764}
```

```
In[7]:= fsincos[x_] = c Sin[x a + b] + d + c Cos[x a + b] /. %
```

```
Out[7]:= 0.366764 - 1.92031 Cos[0.405582 - 1.70351 x] + 1.92031 Sin[0.405582 - 1.70351 x]
```

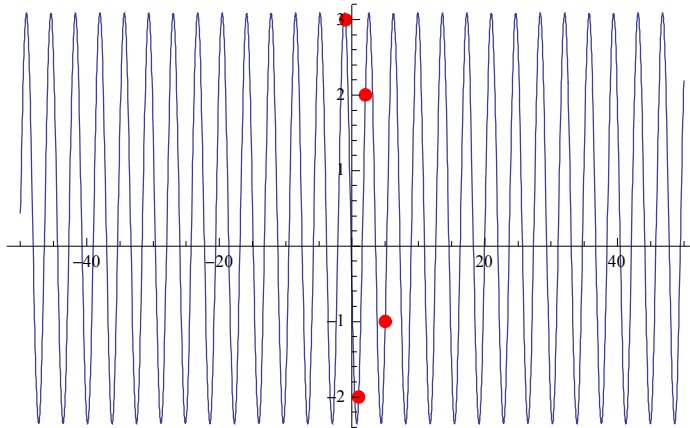
Výsledná funkce byla sestavena pomocí dosazovacího operátoru /. Následuje obrázek nalezené funkce. Intuitivně je zřejmé, že se jedná o periodickou funkci (přesvědčte se!)

```
In[8]:= Show[Plot[fsincos[x], {x, -1, 5}],
  ListPlot[body, PlotStyle -> Directive[Red, PointSize[Large]]]]
```



```
In[9]:= Show[Plot[fsincos[x], {x, -50, 50}],
  ListPlot[body, PlotStyle -> Directive[Red, PointSize[Large]]]]
```

Out[9]=



Ověření, že zadané body leží na nalezené funkci (ostatní body doplňte sami):

```
In[10]:= fsincos[2]
```

Out[10]= 2.

Při experimentování s počátečními podmínkami rovnice lze zjistit, že soustava není určena jednoznačně (nemá jediné řešení). Zde je příklad, který mění koeficienty a a b v nějakém rozumném rozmezí. Pro většinu párů čísel a a b je nalezeno řešení a nalezená funkce prochází danými body. Není-li řešení nalezeno, funkce body mine! Prostudujte si následující obrázek a interpretujte jej.

```
In[11]:= Show[ListPlot[body, PlotStyle -> Directive[Red, PointSize[Large]]],
  Flatten[Table[(res = FindRoot[
    {3 == c Sin[-1 a + b] + d + c Cos[-1 a + b], -2 == c Sin[1 a + b] + d + c Cos[1 a + b],
    2 == c Sin[2 a + b] + d + c Cos[2 a + b], -1 == c Sin[5 a + b] + d + c Cos[5 a + b]},
    {{a, P}, {b, Q}, {c, -1.1}, {d, 0}}, MaxIterations -> 500];
    fsincos[x_] = c Sin[x a + b] + d + c Cos[x a + b] /. res;
    Plot[fsincos[x], {x, -1, 5}]), {P, -2, -1, 1/2}, {Q, -1, 2, 1/3}], 1]]
```

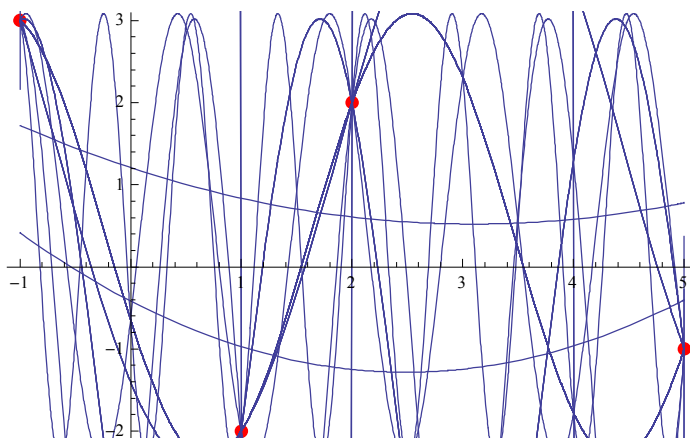
FindRoot::cvmit : Failed to converge to the requested accuracy or precision within 500 iterations. >>

FindRoot::cvmit : Failed to converge to the requested accuracy or precision within 500 iterations. >>

FindRoot::cvmit : Failed to converge to the requested accuracy or precision within 500 iterations. >>

General::stop : Further output of FindRoot::cvmit will be suppressed during this calculation. >>

Out[11]=



Shrnutí

Zde je vidět, že existují i jiné interpolační metody než jen polynom. Je jich k dispozici mnoho druhů (Newtonova, Lagrangeova, Hermitova atd). Jedná se o zajímavou kapitulu numerické matematiky.

Úlohy

- 1/ Zvol si jiné 4 body v rovině a pokus se je interpolovat pomocí funkcí $\sin$ a $\cos$.
 - 1a/ zvol si body na přímce
 - 1b/ zvol si body na hyperbole
 - 1c/ zvol si dva a dva body "daleko" od sebe.
 - 2/ Experimentuj s počátečními podmínkami.
-

Zdroje

<http://reference.wolfram.com/mathematica/ref/FindRoot.html>