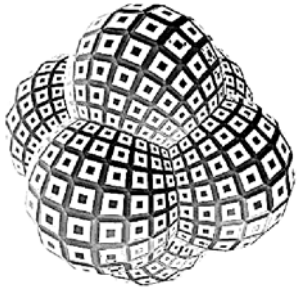


Zborník konferencie o vyučovaní matematiky na VŠ pomocou programového systému firmy

WOLFRAM RESEARCH, INC. USA
MATHEMATICA®

MATHEMATICA '99

Možnosti vizualizácie zobrazení pomocou programového systému *MATHEMATICA®*

Monika Kováčová[1]

Katedra Matematiky, Strojnícka Fakulta STU

Abstrakt: *V článku sa zaoberáme možnosťami definície konečno-rozmerných diferencovateľných zobrazení ako objektov systému MATHEMATICA. Ukážeme možnosti systému MATHEMATICA pri definícii nového objektu a definícii operácií s týmto objektom (skladanie zobrazení, inverzné a identické zobrazenie). Prezentujeme tiež vlastný grafický balík, ktorý umožní pedagógovi jednoduchým spôsobom vizualizáciu zobrazení (napr. pri prenáškach).*

ÚVOD

V súčasnosti môže pedagóg pri príprave výuky využívať veľké množstvo programových systémov a prispôsobiť ich svojim požiadavkám. Väčšina programových systémov ponúka rôzne možnosti vizualizácie funkcií, či už daných parametricky, bodovo, alebo pomocou predpisu. Podľa našich skúseností pri snahe implementovať do programového systému zobrazenia, sa učiteľ stretáva s mnohými problémami. A práve zobrazenia (hoci aj len polárne, alebo sférické) pôsobia často študentom problémy. Štandardne používané zobrazenia v základnom kurze je možné ľahko modelovať a kresliť. Keď však chceme podrobnejšie preskúmať modifikácie týchto zobrazení, prípadne zobrazenia, ktoré nie sú napr. regulárne a prosté, stretávame sa s problémami. Vizualizácia typu tabuľa a krieda neuspokojuje nielen študentov, ale ani učiteľov. V tomto článku prinášame práve ukážku, ako zlepšiť túto situáciu. Predprogramovaním jednoduchého balíka môžeme nielen vizualizovať zobrazenia, ale aj hľadať kompozíciu zobrazení, či inverzné zobrazenie.

Nie je cieľom tohto článku zamýšľať sa nad problémom geometrického videnia a dostatočne, či nedostatočne vyvinutého geometrického cítenia našich študentov. Všetci pedagógovia si pravdepodobne uvedomujú jeho pokles. Dosť obtiažne je naprávať situáciu na vysokej škole. V našom článku sa pokúsime ukázať netradičný prístup k výuke zobrazení.

[1] Mgr. Monika Kováčová, Katedra Matematiky, Strojnícka fakulta STU, Nám. Slobody 17, 812 31 Bratislava, kovacova_v@dekan.sjf.stuba.sk, kovacova@cvt.stuba.sk

Ukážeme, ako je možné vytvoriť nové objekty - v našom prípade objekt zobrazenia a ako definovať funkcie pre prácu so zobrazeniami.

Naším zámerom je pomôcť pri výuke matematickej analýzy v základnom kurze matematiky. V článku sa preto dopúšťame niekoľkých zjednodušení (napr. namiesto presnej definície ľavej a pravej inverzie, ktorá je obvyklá v algebre, definujeme len operáciu inverzie). Každý čitateľ ich však v prípade potreby jednoducho odstráni úpravou nami definovaných funkcií pre prog. systém *MATHEMATICA*.

Zobrazenia

Hlavným problémom prvej časti článku je možnosť definície zobrazenia. Funkcie a objekty, ktoré popíšeme v tejto časti článku budeme využívať pri vytváraní programového balíka v druhej časti. Jeho podrobnú štruktúru nebudeme (vzhľadom na rozsah) uvádzať. V prípade záujmu uľahčí prvá časť článku pochopenie nášho balíka.

Systém *MATHEMATICA* používa niekoľko základných typov premenných. Sú to Symbol, Integer, Real – typy bez ohraničenia dľžky, rovnako ako typ List. Každý objekt má samozrejme svoj presne určený typ (Head [expression]). Keď začneme pracovať so zobrazeniami, situácia je trochu zložitejšia.

Uvažujme zobrazenie f: A $\rightarrow$ B. Tento objekt spája v sebe niekoľko jednoduchších objektov. Pre každé zobrazenie musíme určiť typ jeho definičného oboru, obor hodnôt ako aj formát pravidiel, ktoré toto zobrazenie definujú. Musíme nutne vytvoriť nový objekt typu zobrazenie (mapping) a definovať základné operácie pre tento objekt.

Budeme študovať a popisovať zobrazenia, ktorých definičný obor je podmnožinou konečnorozmerného vektorového priestoru. Existuje niekoľko spôsobov, ako popísať takúto množinu v n-rozmernom priestore, napr. pomocou nerovností. Tieto spôsoby vedú pri spracovaní pomocou systému *MATHEMATICA* ku komplikáciám. Programový systém *MATHEMATICA* neumožňuje bez implementácie dodatočných programových balíkov riešenie nerovníc. Predpokladajme teda, pre jednoduchosť, že definičným oborom a oborom hodnôt našich zobrazení je celý konečnorozmerný priestor. Bod konečnorozmerného priestoru je objekt typu List.

Z programátorského hľadiska na definíciu zob. medzi dvoma konečnorozmernými priestormi musíme určiť okrem dimenzie priestoru definície a priestoru hodnôt aj pravidlo, ktoré bude určovať ako sa bod z priestoru A zobrazí do priestoru B. Teda zobrazenie musí byť v tvare

In[1]:=

```
mapping[oldvariables, rules, newvariables]
```

Out[1]=

```
mapping[oldvariables, rules, newvariables]
```

Objekt tohto tvaru budeme ďalej považovať za zobrazenie. Budeme používať mapping ako novo definovaný typ a objekty tohto typu budeme označovať mapping. Napr.

In[2]:=

```
m=mapping[{x,y},{x^2-y^2,2x y} ,{u,v}]
```

Out[2]=

```
                  2    2
mapping[{x, y}, {x  - y , 2 x y}, {u, v}]
```

bude pre nás označovať zobrazenie z priestoru R^2_{xy} do priestoru R^2_{uv} dané predpisom

$$u = x^2 - y^2 \quad a \quad v = 2 x y.$$

Objekt typu mapping je teda v skutočnosti objektom typu List x List x List, s istými obmedzeniami napr. dĺžka druhého a tretieho List-u musí byť rovnaká, prvý a tretí List musí pozostávať len zo zoznamu premenných.

Keďže pracujeme s novým typom premennej, musíme vytvoriť funkcie, ktoré z tohto typu dokážu extrahovať jednotlivé zložky.

Nazvime ich napr.

dom	funkcia určujúca definičný obor zobrazenia
rules	pravidlá definujúce zobrazenie
cod	funkcia určujúca ko-obor zobrazenia

a definujme ich nasledovne:

In[3]:=

```
dom[map_mapping] := map[[1]];
rules[map_mapping]:=map[[2]];
cod[map_mapping]:=map[[3]];
```

Rovnako je rozumné definovať logickú funkciu, ktorá nám dokáže overiť formálnu správnosť definície nášho zobrazenia

In[6]:=

```
mappingQ[map_mapping]:=
          Length[map]==3 && Depth[dom[map]]==Depth[cod[map]]==2 &&
          Length[rules[map]]==Length[cod[map]];
```

In[7]:=

```
mappingQ[m]
```

Out[7]=

```
True
```

Nové zobrazenie môžeme vytvoriť pomocou skladania zobrazení, alebo pomocou operácie inverzie. Dôležitý je aj test rovnosti zobrazení.

Zhodnosť zobrazení

Definujme test rovnosti zobrazení. Potrebujeme zistiť, či dva objekty typu mapping sú totožné. Jedna z možností je:

In[8]:=

```
zhodaQ[f1_mapping,f2_mapping]:=
    (dom[f1]==dom[f2]) && (cod[f1]==cod[f2]) &&
    (Simplify [rules[f1]-rules[f2]]==Table[0,{length[cod[f1]]} ]);
```

ZhodaQ je operácia, ktorá vráti objekt typu Boole (True, False) a je teda typu

$$\text{mapping x mapping} \rightarrow \text{Boole}$$

Skladanie zobrazení

Zložené zobrazenie (ak existuje) je objekt typu mapping x mapping $\rightarrow$ mapping

In[9]:=

```
skladanie[f_mapping,g_mapping]:= mapping[dom[f],rules[g] /. Thread[cod
[f] -> rules[f]],cod[g]] /; cod[f]===dom[g] ;
```

Použime nami definovanú operáciu skladania na jednoduchom príklade. Ak

In[10,11]:=

```
f=mapping[{x,y},{x^2 +y^2,-2x y},{u,v}]
g=mapping[{u,v},{u+v,u-v},{r,s}]
```

Out[10,11]=

```
                    2    2
mapping[{x, y}, {x  + y , -2 x y}, {u, v}]
mapping[{u, v}, {u + v, u - v}, {r, s}]
```

potom

In[12,13,14]:=

```
rules[g]
Thread[cod[f]->rules[f]]
rules[g] /.Thread[cod[f]->rules[f]]
```

Out[12,13,14]=

```
{u + v, u - v}
        2    2
{u -> x  + y , v -> -2 x y}
  2           2   2           2
{x  - 2 x y + y , x  + 2 x y + y }
```

In[15]:=

```
skladanie[f,g]
```

Out[15]=

```
                   2            2   2            2
mapping[{x, y}, {x  - 2 x y + y , x  + 2 x y + y }, {r, s}]
```

Môžeme overiť, že výsledné zobrazenie je opäť zobrazením v zmysle našej definície

In[16]:=

```
mappingQ[skladanie[f,g]]
```

Out[16]=

```
True
```

Identické zobrazenie

Identické zobrazenie je operácia typu List→ mapping

In[17]:=

```
identityMapping[var_List]:= mapping[var,var,var]
```

In[18]:=

```
identityMapping[{x,y}]
```

Out[18]=

```
mapping[{x, y}, {x, y}, {x, y}]
```

Inverzné zobrazenie

Inverzné zobrazenie (ak existuje) je objekt typu mapping. Funkcia inversemap musí byť objekt typu mapping → mapping. Pri hľadaní inverzného zobrazenia môže existovať niekoľko rôznych inverzných pravidiel určujúcich inverzné zobrazenie (v závislosti od definičného oboru), typ našej novej funkcie môže byť aj mapping→ ListOf Mappings.

Jedna z možností definície je:

In[19]:=

```
inversemap[f_mapping]:=
With[
{riesenie = Simplify[PowerExpand[Simplify[
    Solve[Thread[rules[f]==cod[f]],dom[f]]]]]},
Map[mapping[cod[f],#,dom[f]]&,
    dom[f]/.riesenie]];
```

Funkčnosť takto defnovaného príkazu overme na jednoduchom príklade :

Nájdite inverzné zobrazenie ku zobrazeniu *f:* $R^2_{xy} \to R^2_{uv}$ definovanému predpisom

$$u = 2x - y,\ v = x + 2y$$

In[20]:=

```
f=mapping[{x,y},{2x-y,x+2y},{u,v}];
```

In[21]:=

```
inversemap[f]
```

Out[21]=

```
                    2 u + v  -u + 2 v
{mapping[{u, v}, {-------, --------}, {x, y}]}
                      5        5
```

Vizualizácia diferencovateľného zobrazenia

V tejto časti článku predvedieme niektoré možnosti nami definovaného príkazu **mapGraphics**. Pomocou neho môžeme študentom demonštrovať nielen funkcie, ale aj tie zobrazenia, ktorých definičný obor má dimenziu 1, alebo 2 a obor hodnôt dimenziu 2, 3 (pričom nemusia byť nutne regulárne).

Jeho vytvorenie nie je zložité a kompletný modul prezentovaný v práci, je dostupný u autora. Použitím príkazov definovaných v prvej časti nášho článku môžeme tiež vizualizovať skladanie zobrazení alebo operáciu inverzie.

In[22]:=

```
<<Geometry`mapGraphics`
```

mapGraphics[mapping,"meno", range[s]]

kde range[s] je buď interval v tvare {a,b}, alebo dvojica intervalov , aj s určením veľkosti kroku v tvare $\{a_1, a_2, step_a\}$, $\{b_1, b_2, step_b\}$

Tento príkaz vytvorí objekt typu `Graphics`. To už je štandardný objekt, ktorý dokážeme prezerať napr. pomocou príkazu **`Show`**. Použitie tohto príkazu si ukážeme na niekoľkých príkladoch. (Zaradili sme aj zobrazenie, ktoré nie je prosté.)

In[23]:=

```
map1=mapping[{t},{t^2,t^3},{x,y}];
map2=mapping[{t},{Sin[t],Cos[t],Sin[t]^2},{x,y,z}];
map3=mapping[{x,y},{x-y,2x-2y},{u,v}];
map4=mapping[{x,y},{2x-y,x+2y},{u,v}];
```

Ukážeme si aj zobrazenia z 2-dimenzionálneho priestoru do 2 a 3 dimenzionálneho priestoru

In[27]:=

```
map5=mapping[{r,t},{r Cos[t],r Sin[t]},{x,y}];
map6=mapping[{t,u},{Cos[t],Sin[t],u},{x,y,z}];
map7=mapping[{x,y},{x^2+y^2,-2x y},{u,v}];
map8=mapping[{u,v},{Cos[v]*Cos[u],Cos[v]*Sin[u],Sin[v]},{x,y,z}];
```

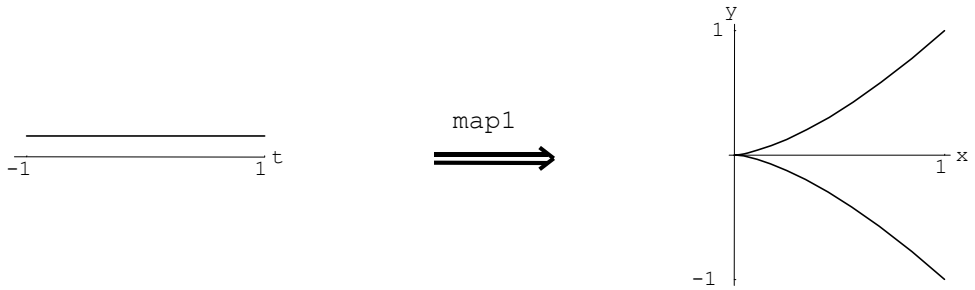
Pre tieto zobrazenia dostaneme nasledovné grafické výstupy

In[31]:=

```
mapGraphics[map1,"map1",{-1,1}];
```

In[32]:=

```
Show[%]
```

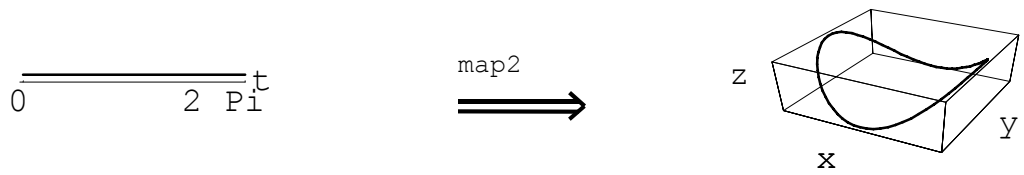


Out[32]=

```
-Graphics-
```

In[33]:=

```
Show[mapGraphics[map2,"map2",{0,2 Pi}]]
```

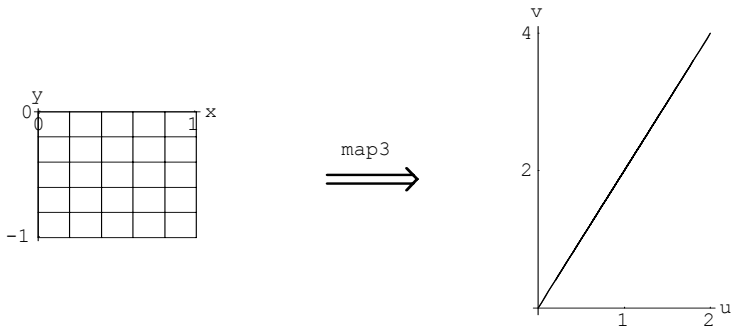


Out[33]=

```
-Graphics-
```

In[34]:=

```
Show[mapGraphics[map3,"map3",{0,1,0.2},{-1,0,0.2}]]
```



Out[34]=

```
-Graphics-
```

In[35]:=

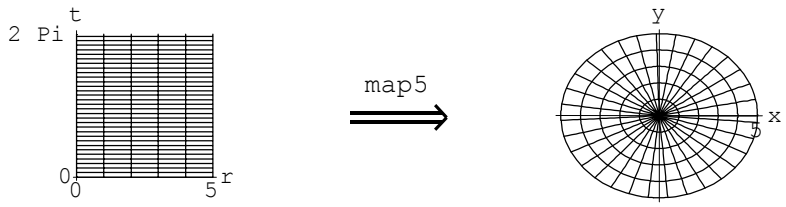
```
Show[mapGraphics[map4,"map4",{0,1,0.2},{0,1,0.2}]]
```

Out[35]=

```
-Graphics-
```

In[36]:=

```
Show[mapGraphics[map5,"map5",{0,5,1},{0,2Pi,0.2}]]
```



Out[36]=

```
-Graphics-
```

In[37]:=

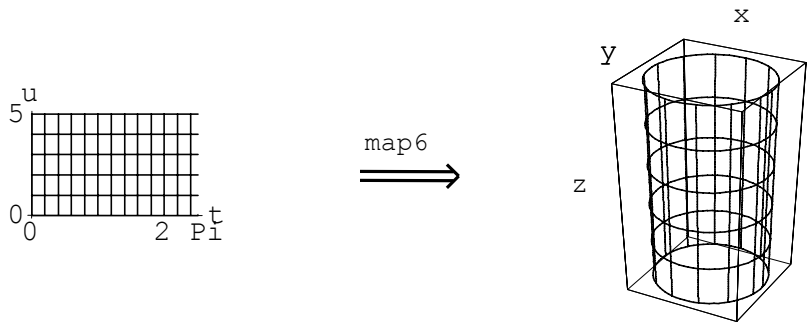
```
Show[mapGraphics[map5,"map5",{0,5,1},{0,Pi,Pi/2}]]
```

Out[37]=

-Graphics-

In[38]:=

```
Show[mapGraphics[map6,"map6",{0,2Pi,0.5},{0,5,1}]]
```

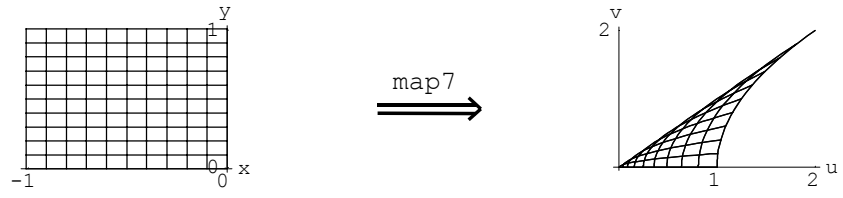


Out[38]=

-Graphics-

In[39]:=

```
Show[mapGraphics[map7,"map7",{-1,0,0.1},{0,1,0.1}]]
```

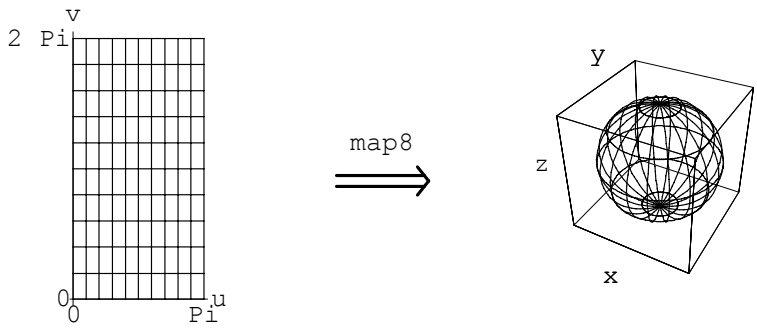


Out[39]=

-Graphics-

In[40]:=

```
Show[mapGraphics[map8,"map8",{0,Pi,Pi/10},{0,2Pi,Pi/5}]]
```



Out[40]=

```
-Graphics-
```

Uveďme ešte ukážku príkladu demonštrovateľnú na prednáške alebo na cvičení

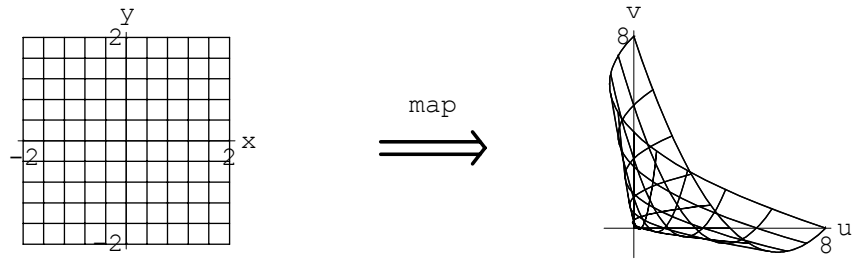
Úloha: nájdite inverzné zobzanie k zobrazeniu map a overte správnosť výsledku

In[41]:=

```
map=mapping[{x,y},{x^2-x y, x y+y^2},{u,v}];
```

In[42]:=

```
Show[mapGraphics[map,"map", {-2,2,0.4},{-2,2,0.4}]]
```



Out[42]=

```
-Graphics-
```

In[43]:=

```
inverse=inversemap[map];
```

Tvar inverznej funkcie priamo v tomto článku nebudeme uvádzať, je príliš dlhý. Správnosť výsledku možeme overiť kompozíciou zobrazenia a zobrazenia k nemu inverznému. *MATHEMATICA* , ako výsledok procedúry inverse našla 4 rôzne zobrazenia. Len druhé

z týchto zobrazení je pravou inverziou, ostatné sú ľavými inverziami. Výsledok kompozície vidíme v nasledujúcom výstupe.

In[44]:=

```
Map[composition[map,#]&,inverse]//PowerExpand//Simplify
```

Out[44]=

```
{mapping[{x, y}, {-x, -y}, {x, y}],

 mapping[{x, y}, {x, y}, {x, y}],

                      x - y     x + y
 mapping[{x, y}, {-(-------), -------}, {x, y}],
                     Sqrt[2]   Sqrt[2]

                    x - y      x + y
 mapping[{x, y}, {-------, -(-------)}, {x, y}]}
                   Sqrt[2]    Sqrt[2]
```

Ako sme spomínali už v úvode nášho článku, neobsahuje tento príspevok všetky možné definície pre prácu so zobrazeniami. Ak by sme ho chceli použiť pre výuku základov algebry, je nutné predefinovať nielen operáciu inverzie na operáciu závislú od poradia, ale definovať aj napr. test injektívnosti, surjektívnosti resp. bijektívnosti zobrazenia.

Myslíme si však, že uvedená definícia objektu mapping je ”rozumne” zvolená a čitateľ v prípade záujmu na tu uvedených základoch dokáže potrebné funkcie, resp. moduly vytvoriť.

Literatúra:

[1] Ďurikovičová M.: *Metrické úlohy v priestore,* ***MATHEMATICA* 99**, Bratislava 29.6.-2.7. 1999, pp.

[2] Halada L.: *Použitie systému Dotest pri výčbe a skúšaní matematiky,* ***MATHEMATICA* 99**, Bratislava 29.6.-2.7. 1999, pp.

[3] Kolesárová A., Kováčová M., Záhonová V.: *Použitie programového systému Mathematica pri výučbe základov matematickej analýzy na Sjf STU,* Matematická štatistika a Numerická matematika, Kálnica 1. -5. júna 1998 , pp.134-141

[4] Kolesárová A.: *Výučba Fourierových radov s podporou programového systému Mathematica,* ***MATHEMATICA* 99,** Bratislava 29.6.-2.7. 1999, pp.

[5] Kováčová M.: *Výučba diferenciálneho počtu funkcie viac premenných pomocou pg. systému Mathematica,* Zborník 25. konferencia VŠTEZ - Matematika v inžinierskom vzdelávaní, Trnava 7. - 10. sept. 1998, pp. 173-180

[6] Omachelová M.: *Zisťovanie priebehu funkcie jednej reálnej premennej s podporou pg. systému MATHEMATICA ,* ***MATHEMATICA* 99**, Bratislava 29.6.-2.7. 1999, pp.

[7] Záhonová V.: *Lineárne diferenciálne rovnice n-tého rádu s konštantnými koeficientami a program. systém MATHEMATICA ,* ***MATHEMATICA* 99,** Bratislava 29.6.-2.7. 1999, pp.

[8] Wolfram Research: *The Mathematica Book 3rd ed.*,Wolfram Media/Cambridge University Press, 1996.