

# VIZUALIZACE 2D VEKTOROVÝCH DAT

*Tomáš Stolín*

Katedra informatiky, Ostravská Univerzita v Ostravě

## 1 Úvod

Častou úlohou vizualizace dat je zobrazení veličiny, resp. několika veličin v kontinuálním prostředí. Např. geografická mapa zobrazuje skalární hodnotu - nadmořskou výšku povrchu. Přitom pro vizualizaci se zpravidla používá buď systém izolinií (vrstevnic), nebo pro lepší orientaci se každé zobrazované hodnotě přiřadí barva, z předem zvolené barevné škály.

Jelikož barvu na výstupních zařízeních vyjadřujeme na základě tří základních barev, nabízí se možnost současného zobrazování tří veličin v každém bodě. V příspěvku je analyzován vztah barevného modelu (RGB, HSV) a zobrazování multiskalárních resp. vektorových dat. Dále je ukázána možnost využití programového prostředí MATLAB k vizualizaci právě uvedených dat.

## 2 Skalární data

Metodika zobrazování skalárního pole je jedna z nejpropracovanějších a nejčastěji zmiňovaných metodik vizualizace dat. Skalár je fyzikální veličina, která má pouze velikost. Potřebujeme tedy zobrazit jen jednu informaci. Z různých metod vyjmenuji nejčastěji používané – přímé zobrazování hodnot v rovině, izolinie a především barevné kódování (více viz např.[1]).

Je důležité si uvědomit, že k zobrazování skalárů pomocí barvy nám stačí pouze odstíny šedi, popřípadě jiné jediné barvy. Je to dáno tím, že skaláry ve své podstatě nenabývají záporných hodnot, a proto není třeba zdůraznit nulový bod, jak je tomu u vektorů.

## 3 Multiskalární a vektorová data

U multiskalárních a vektorových dat je tedy potřeba zobrazit několik informací v jednom bodě (případně v malé oblasti). Ač se tato data někdy formálně zapisují stejně jako uspořádané  $n$ -tice hodnot, je nutné vzít na zřetel podstatný rozdíl mezi nimi. V případě multiskalárních dat zobrazujeme v jednom bodě několik veličin, u nichž není uvažována vzájemná vnitřní vazba. U vektorových dat má smysl uvažovat např. velikost, směr, atd., tj. sdruženou charakteristiku jednotlivých komponent.

Metod k vizualizaci multiskalárních dat není mnoho. Pravděpodobně nejčastěji používanou technikou je kombinace několika technik pro zobrazování skalárních dat, kdy je každá složka multiskalárních dat zobrazena jinou technikou. Další technikou jsou např. glyfy [1, 4]. Při kódování čistě pomocí barvy jsme schopni zobrazit maximálně tři informace. Např. každou ze tří veličin můžeme mapovat na jednu z hran kostky modelu RGB. V případě 3D vektorového pole se tímto postupem nulový bod zobrazí 50% šedou barvou. Body blízko nuly pak nejsme schopni odlišit. Proto přímá vizualizace 3D vektorů pomocí barvy je diskutabilní.

Téměř jedinou používanou metodou vizualizace vektorového pole je reprezentace vektorů pomocí šipek. V dalším textu však budou popsány techniky, které při zobrazování 2D vektorového pole používají barvy. Na základě výsledků vizualizace vektorů pomocí těchto technik později vyhodnotíme přínos barvy při kódování vektorových dat.

## 4 Vizualizace vektorových dat

### 4.1 Klasická technika – šipky

Standardní technika, implementována snad ve všech matematických a vizualizačních programech, je zobrazování vektorů pomocí šipek (popř. malých úseček či jehliček) – viz obr.3. Šipka je symbolem, který umí zobrazit minimálně dvě informace; první typ informace (např. velikost vektoru) je vyjádřen délkou šipky a další typ (např. směr vektoru) je vyjádřen sklonem šipky. Další informaci lze vyjádřit např. vhodným umístěním počátku šipky v prostoru, barvou šipky a tloušťkou „těla“ šipky.

Jednou z nevýhod této techniky je, že pokud jsou zobrazovány dlouhé vektory zasahující svým zobrazením do jiných vektorů, obrázek se stává nepřehledným a těžko čitelným. Toto lze vyřešit zkrácením všech vektorů, ale v tomto případě nastává problém opačný – malé vektory se stanou po zkrácení tak malými, že se buď vůbec nezobrazí a nebo budou špatně viditelné. V obou případech ztrácí vizualizace význam.

### 4.2 Barevná technika

Jedná se o techniku, která byla publikována v [2]. Byla navržena tak, aby dokázala zobrazit celé vektorové pole (popřípadě jeho určitou oblast) najednou. Musí tedy kódovat informaci o každém vektoru, tzn. jeho počátek, směr a velikost, tak, aby výsledná vizualizace vektoru zabrala na obrazovce co nejméně místa, v ideálním případě velikost jednoho pixelu. **Pomocí barvy dokážeme zobrazit dvě informace v jednom bodě** (směr a velikost vektoru), zbývající informaci (počátek vektoru) vyjádříme polohou zobrazovaného barevného bodu v souřadnicovém systému na obrazovce.

Jakým způsobem se vektorům přiřazuje výsledná barva je popsáno v následujících odstavcích.

Je-li vektorové pole  $\mathbf{u}$  omezené, tzn. platí-li pro všechny vektory, že jejich velikost je shora omezená konstantou  $\rho$ , pak prostorem, který vymezuje hodnoty vektorů, je **kruh se středem v počátku kartézské soustavy souřadnic a s poloměrem  $\rho$** . Barevným prostorem, který bude použit pro kódování vektorů, je tedy kruh s výše uvedenými vlastnostmi. **Rozložení barev na ploše kruhu odpovídá rozložení barev na podstavě kuželů barevného modelu HLS** (viz obr. 5). Černá (popřípadě bílá) barva je určena pro nedefinované hodnoty.

Zaveďme nyní polární reprezentaci vektorového pole  $\mathbf{u}$ . Tzn., že

$$\mathbf{u}(x, y) = (v(x, y), w(x, y)) = (\varphi, \rho), \text{ kde}$$
$$\varphi = \arctan \frac{w(x, y)}{v(x, y)} \text{ a } \rho = \sqrt{v(x, y)^2 + w(x, y)^2}$$

Jednoznačné přiřazení barvy pro jednotlivé vektory se potom provede ve dvou krocích:

1. Určení barevného tónu – barevný tón je přiřazen vektoru podle jeho směru  $\varphi$ , který lze chápat jako parametr barevného tónu (hue) modelu HSV
2. Určení výsledné barvy – sytost barvy je úměrná velikosti vektoru  $\rho$ . Konkrétně velikost parametru sytosti (saturation) modelu HSV bude rovna poměru  $\rho / r$ . V případě, že délka vektoru  $\rho$  je větší než poloměr  $r$ , je parametr jasu roven 0 (černá barva), jinak je jas roven 1 (pracujeme s barvami podstavy kužele modelu HSV).

### 4.3 Kombinovaná technika

Tato technika byla navržena tak, aby co nejlépe řešila nedostatky předcházejících dvou metod (viz odstavec 4.1 a 6) a zároveň využila všech předností barevného kódování, především možnosti zobrazení informace v jednom pixelu (popř. v oblasti  $n \times n$  pixelů, kde  $n$  je malé celé číslo).

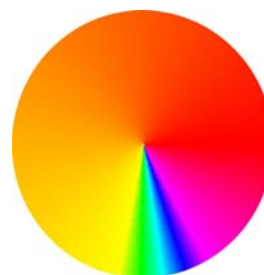
**Pomocí barvy je kódována pouze jedna informace, a sice směr vektoru. Druhá informace, délka vektoru, je znázorněna pomocí vrstevnic.**

Protože je barvou kódována jen jedna informace, vystačí nám pouze čisté barvy, které se nacházejí na obvodu podstavy kužele reprezentujícího model HSV (viz obr.2, na kterém je tento model upraven dle potřeb uživatele; konkrétně se vztahuje k obr.7). Princip přiřazení barvy vektoru je analogický s přiřazením barevného tónu danému vektoru u barevné techniky (viz odstavec 4.2). Výjimku tvoří pouze nulový vektor, kterému je přiřazena bílá barva. Černá barva je vyhrazena pro nedefinované hodnoty.

Tato metoda je kombinací barevné techniky a vrstevnic, proto bude v dalším textu označována jako **kombinovaná technika**.



Obr.1: Podstava kuželů barevného modelu HLS



Obr.2: Upravený barevný model HLS dle potřeb uživatele; pro přehlednost je čistou barvou vykreslena vždy i spojnice ze středem

## 5 Implementace technik

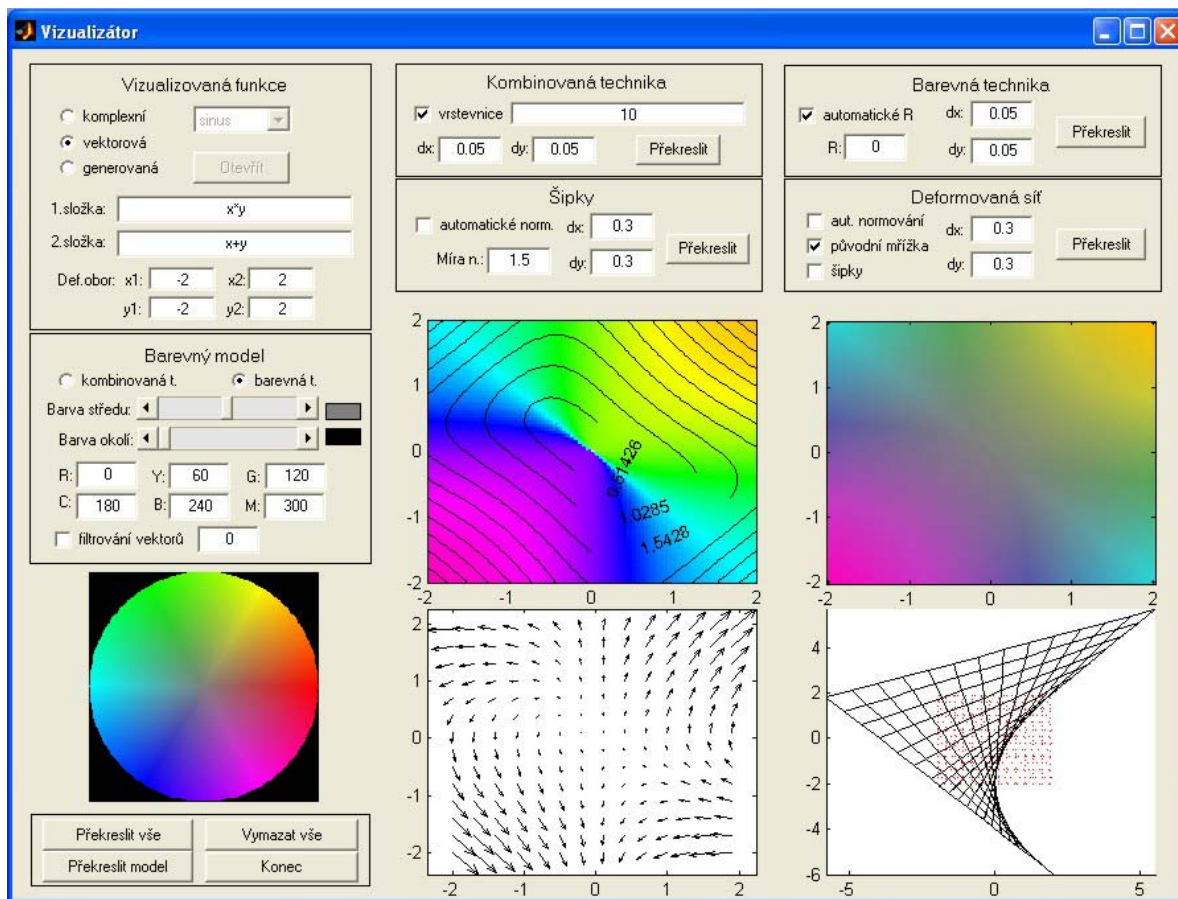
Implementace vizualizačních technik byla provedena v programovém prostředí MATLAB 6.5 Release 13. MATLAB byl zvolen především proto, že již obsahuje širokou škálu vizualizačních funkcí, které bylo možno využít.

K zobrazení vektorových dat pomocí šipek byla použita vestavěná funkce *quiver*, pro implementaci technik používajících ke kódování informací barvu byly využity především funkce *colormap* a *pcolor*. Zjednodušeně řečeno - pomocí první z nich lze nastavit barevnou paletu dle našich potřeb a pomocí druhé pak vykreslit barevnou plochu podle nastavené palety. Pro vykreslení vrstevnic byla použita funkce *contour*.

V tomto prostředí byla dále s využitím prostředků GUIDE vytvořena aplikace Vizualizátor (obr.3), která dovede zobrazovat různá 2D vektorová data a komplexní funkce s využitím výše zmíněných technik.

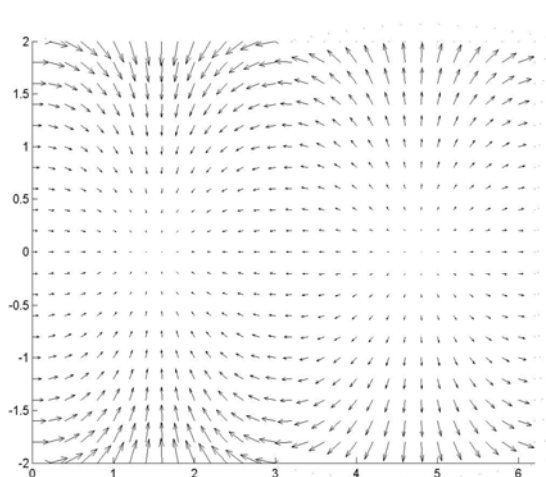
## 6 Příklady vizualizace

V tomto odstavci jsou zobrazeny vizualizace komplexní funkce  $w = \cos(z)$ , pomocí výše zmíněných technik (obr.4-6), a dále ukázka možnosti využití práce s přizpůsobeným barevným modelem dle vlastních potřeb (obr.7). Všechny vizualizace byly provedeny v programovém prostředí MATLAB 6.5. Komplexní funkce byla zvolena proto, že je vhodným generátorem 2D vektorových dat (každé komplexní číslo lze chápat jako uspořádanou dvojici, která je geometricky reprezentovatelná vektorem). Definici této komplexní funkce lze nalézt např. v [3].

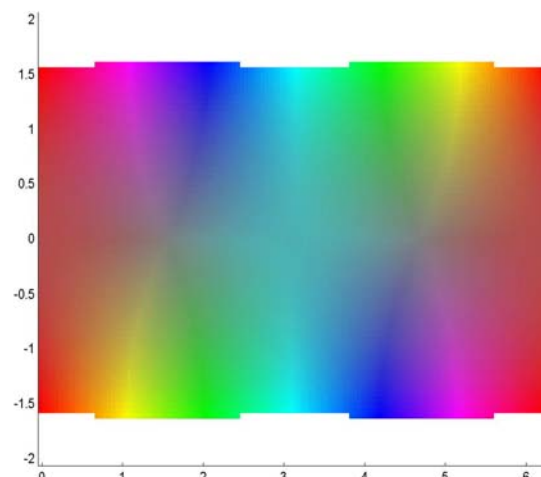


Obr. 3: Hlavní okno programu Vizualizátor

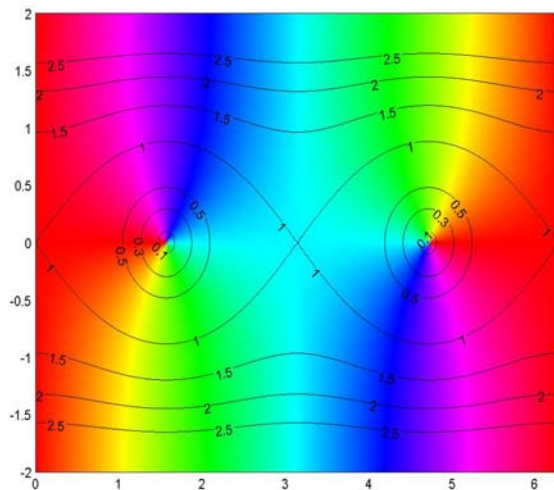
Na obr.7 je zobrazen příklad z praxe – jde o výstup 2D vektorového pole posunutí testovací úlohy pružnosti (více viz [7]). Protože mají všechny vektory směr zhruba z intervalu  $\langle 180^\circ, 300^\circ \rangle$ , je využití základního barevného modelu nevhodné, jelikož by byla použita mála škála barev. Proto je barevný model příslušně upraven tak, aby zmíněný interval pokrývala širší škála barev (obr.2).



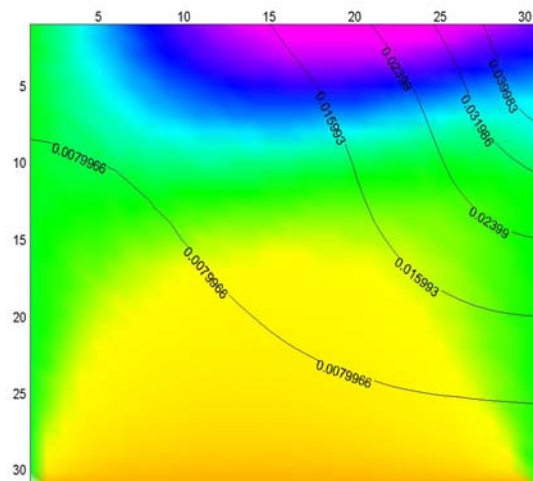
Obr.4: Kosinus – šípky



Obr.5: Kosinus – barevná technika



Obr.6: Kosinus – kombinovaná technika



Obr.7: Simulace odkrytého základu – kombinovaná technika s upraveným modelem

## 7 Závěr

V příspěvku byl navržen formální aparát pro zobrazování vícerozměrných dat s využitím barvy. Na základě provedených vizualizací a porovnání jednotlivých technik jednoznačně vyplývá, že ***použití barevného kódování informací při zobrazování dvourozměrných vektorových dat je přínosem.***

Při kódování dvou informací pomocí barvy však vystává jeden problém. Lidským okem jsou těžko rozlišitelné barvy se stejným barevným tónem, u kterých se mění pouze sytost, a pak také barvy ze stupnice šedi (viz obr.4). Toto nastává především v situacích, kdy velikost vektorového pole není omezená. Právě tehdy je totiž nutné interpolovat v intervalu sytosti  $\langle 0, 1 \rangle$  velké množství hodnot, z čehož vyplývají výše zmíněné potíže. Pro větší přehlednost výsledné vizualizace je pak nutné vektorové pole omezit vhodnou konstantou s tím, že všechny vektory s velikostí rovnou nebo větší než tato konstanta budou zobrazeny s maximální sytostí (tedy bílou barvou; viz „schody“ na obr.4).

V případě barevné reprezentace jedné informace si lze vystačit z čistými barvami, které se nacházejí na obvodu podstavy kuželů barevného modelu HLS. Rozlišení takových barev už většině osob nedělá potíže. Navíc si může každý jedinec přizpůsobit barevný model svým potřebám a schopnostem rozlišování barev (obr.7). Zbývající informaci je pak nutno zobrazit jinou technikou, v případě kombinované techniky pomocí izolinií.

Lze namítnout, jestli barevné kódování pouze jedné informace není zbytečné plýtvání „barevným potenciálem“, když barvou lze teoreticky kódovat až tři informace (např. u modelu HLS se při kódování vůbec nevyužívá světlost barev, ale pouze sytost a barevný tón). Domnívám se, že, alespoň u vizualizace vektorového pole, tomu tak není. Směr vektoru je sice skalární veličina, avšak díky své „periodicitě“ se od ostatních výrazně odlišuje. Pokud je tedy cílem přehledné a srozumitelné zobrazení zdrojových dat (jako na obr.6 a 7), pak se barevné kódování jedné informace pomocí celé barevné škály vyplatí.

A na závěr poznámka (nejen) pro čtenáře tištěného černobílého sborníku – všechny uvedené příklady vizualizace, a mnohé další, lze nalézt na [www](http://www) stránkách uvedených v kontaktních informacích na konci tohoto příspěvku.

## 8 Reference

- [1] FOSDICK, L. D., JESSUP, E. R., et. al.: *An Introduction to High-Performance Scientific Computing*. The MIT Press, London 1995.
- [2] MITZOVÁ, M.: *Vizualizace vícerozměrných dat*. Diplomová práce, OU. Ostrava 2001.
- [3] PRIVALOV, I.I.: *Analytické funkce*. Nakladatelství ČSAV, Praha 1955.
- [4] ŽÁRA, J., BENEŠ, B., FELKEL, P.: *Moderní počítačová grafika*. Computer Press, Praha 1998.
- [5] COSSU, R.: *Coloured visualization for numerical modelling. WSCG'2001*. The 9<sup>th</sup> International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2001, (V.Skala, ed.). ZČU Plzeň 2001.
- [6] BLAHETA, R., et. al.: *GEM22 – programová dokumentace*. ÚGN AVČR, Ostrava 1995.
- [7] STOLÍN, T.: *Vizualizace vícerozměrných dat*. Diplomová práce, OU. Ostrava 2004

## Kontaktní informace

Mgr. Tomáš Stolín  
Katedra informatiky, Přírodovědecká fakulta  
Ostravská Univerzita v Ostravě

Email: [tomstolin@centrum.cz](mailto:tomstolin@centrum.cz)

Mobil: 603948318

www: <http://tomstolin.wz.cz>