

VYUŽITÍ MATEMATICKÉHO PROGRAMU SAGE V ANALÝZE REDISTRIBUČNÍCH SYSTÉMŮ

MUŽÍKOVÁ, Karolína, (SK)

Abstrakt. Matematický program Sage je volně dostupný software, který je určený pro matematické výpočty a grafická znázornění. Pomocí grafického rozhraní v internetovém prohlížeči je nabízeno interaktivní grafické uživatelské prostředí, ve kterém lze s využitím programovacího jazyka Python řešit řadu matematických úloh. Hlavním námětem této práce je rozbor fungování a aplikace programu Sage při analýze redistribučních systémů. Teorie redistribučních systémů spadá pod oblast teorie her a zkoumá rozdělování výplat v systémech s více hráči. Práce je zaměřena na rozbor redistribuce výplat v systému se třemi hráči, což lze s využitím programu Sage přehledně znázornit v trojrozměrném prostoru.

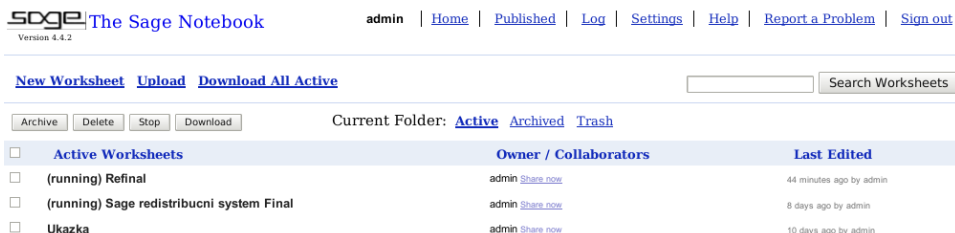
1 Úvod

Tento článek seznamuje s matematickým softwarem Sage, který je volně šiřitelný pod všeobecnou veřejnou licenci GNU (GNU General Public License). Vývoj tohoto open source programu probíhá od roku 2005. Jde o software, který lze výhodně využít při matematických výpočtech a jejich grafickém znázornění. Představuje tak konkurenci drahým komerčním programům jako je například Matlab, Maple a Mathematica [2]. V současné době je k dispozici pro většinu platforem včetně Linux nebo Microsoft Windows. S programem Sage lze pracovat buď v textovém okně terminálu nebo používat interaktivní grafické rozhraní zobrazené v internetovém prohlížeči. Program Sage je tvořený v programovacím jazyce Python. Jde o dynamický, interaktivní, objektově-orientovaný programovací jazyk, který byl prvně publikován v roce 1991. Python je vyvíjen jako open-source projekt, stejně tak jako samotný Sage [4].

2 Základy práce v matematickém programu Sage

Při zpracovávání tohoto článku bylo používáno grafické rozhraní programu Sage zobrazené v internetovém prohlížeči (příkazem `sage -notebook` v okně terminálu). Použita byla

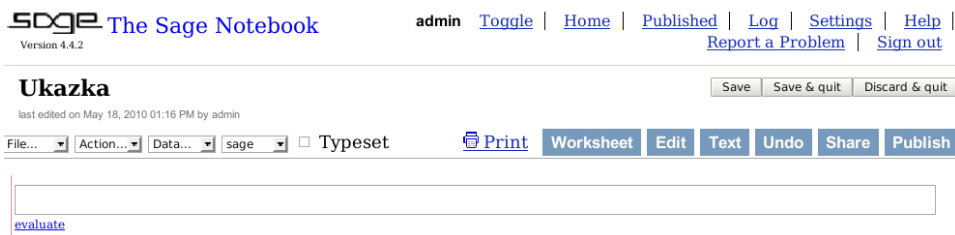
verze 4.4.2. Úvodní uživatelská nabídka po přihlášení do programu je ukázána na Obr. 1. Úvodní strana nabízí možnost výběru existujících pracovních sešitů (Active Worksheets)



Obrázek 1: Úvodní strana programu Sage

nebo založení nového sešitu (New Worksheet). Celková nabídka je uživatelsky velmi přehledná a nabízí standardní volby (ukládání a nahrávání souborů, odstraňování souborů, přejmenování názvů souborů, převod souborů do textové podoby, možnosti sdílení souborů, aj.).

V pracovním sešitě (Obr. 2) se příkazy zadávají do jednotlivých fialových obdélníkových polí. V každé buňce může být i několik příkazů zároveň. Je možné psát každý nový příkaz na nový řádek dané buňky nebo jednotlivé příkazy na jednom řádku oddělovat středníkem. Hotový příkaz se spustí volbou evaluate nebo kombinací kláves `Shift + Enter`. Tím se pod polem s příkazem zobrazí buď výsledek konkrétního výpočtu nebo se vygeneruje obrázek. Pro ukládání se používá příkaz `Save` v pravé horní liště, případně příkaz `Save & quit`, který



Obrázek 2: Nový pracovní sešit

konkrétní pracovní sešit uloží a uzavře. Při ukončení práce je nutné odhlášení z programu Sage (Sign out) a následné použití kláves `Ctrl + C` v terminálu.

Program Sage nabízí velmi podrobnou nápovědu, a to v různých podobách. Uživatelsky nejvhodnější a nejprehlednější je Sage Tutorial a Sage's Reference Manual, které obsahují všeobecné informace o jednotlivých funkcích a velké množství příkladů [6]. V samotném programu lze využít znak otazníku umístěný za nějaký název objektu, který vypíše o daném objektu základní informace, včetně informace o typu funkce, její syntaxi a uvádí příklad použití. Příklad fungování této nápovědy je ukázán na Obr. 3 pro funkci `implicit_plot3d`, která byla v analýze redistribučního systému také použita.

```

implicit_plot3d?

File: /usr/local/sage/local/lib/python2.6/site-packages/sage/plot/plot3d/implicit_plot3d.py
Type: <type 'function'>
Definition: implicit_plot3d(f, xrange, yrange, zrange, **kwargs)
Docstring:
Plots an isosurface of a function.
INPUT:
  • f - function
  • xrange - a 2-tuple (x_min, x_max) or a 3-tuple (x, x_min, x_max)
  • yrange - a 2-tuple (y_min, y_max) or a 3-tuple (y, y_min, y_max)
  • zrange - a 2-tuple (z_min, z_max) or a 3-tuple (z, z_min, z_max)
  • plot_points - (default: "automatic", which is 50) the number of function evaluations in each direction. (The number of cubes in the marching cubes algorithm)
  • contour - (default: 0) plot the isosurface f(x,y,z)=contour. Can be a list, in which case multiple contours are plotted.
  • region - (default: None) if region is given, it must be a Python callable. Only segments of the surface where region(x,y,z) returns a number >0 will be included
EXAMPLES:
sage: var('x,y,z')
sage: (x, y, z)
A simple sphere:
sage: implicit_plot3d(x^2+y^2+z^2==4, (x, -3, 3), (y, -3,3), (z, -3,3))

```

Obrázek 3: Návod k funkci `implicit_plot3d`

3 Teorie redistribučních systémů

Teorie redistribučních systémů je aplikací a rozšířením oblasti teorie her. Redistribuční systém je takový systém, ve kterém dochází k nějakému přerozdělení odměn oproti výkonům, které podali členové tohoto systému. Pokud dochází v systému k redistribuci prostředků mezi členy, kteří tento systém vytvářejí, a tato redistribuce neodpovídá výkonům, které jednotliví členové vykonali, snižuje se tím výkon celého systému. Výzkumu této oblasti se velmi intenzivně věnuje vědecký kolektiv na Vysoké škole finanční a správní v Praze [1].

Model elementárního redistribučního systému, jak ho navrhuje Valenčík a Budinský [1], zahrnuje pouze tři hráče A, B, C. Každý z těchto hráčů má stejný vliv na daný systém, libovolní dva hráči mohou vytvořit koalici, všechny koalice jsou možné a rovnoprávné a všichni hráči jsou informováni o tom, jaká je jejich výkonnost. Výkonnost hráčů A, B, C je ve stejném pořadí oceněna hodnotami 6 :4:2 (jednotka není pro účel zkoumání redistribučního systému exaktně daná, může jít např. o tisíce peněžních jednotek).

Tento základní model popisuje redistribuční rovnice, která vyjadřuje snížení efektivnosti systému v důsledku odchylky výplat od výkonnosti. Rovnice popisující všechny možnosti rozdělení výplat je

$$x + y + z = 12 - \eta \sqrt{(x - 6)^2 + (y - 4)^2 + (z - 2)^2} \quad (1)$$

kde $x + y + z$ představuje součet skutečných výplat, hodnota 12 vyjadřuje maximální možnou výplatu (6 + 4 + 2), koeficient η je koeficient snížení výkonnosti systému (používá se hodnota $\eta = 0.5$; pokud je $\eta = 0$, tak redistribucí nedochází ke snížení efektivnosti).

Výraz pod druhou odmocninou představuje funkci prostorové vzdálenosti rozdělení skutečných výplat od výplat podle výkonnosti hráčů.

Uvedenou elementární redistribuční rovnici lze znázornit v trojrozměrném prostoru. Vznikne tím redistribuční plocha. Na této ploše leží dva významné body, a to bod V rozdělení

výplat podle výkonu (s hodnotou souřadnic $[6, 4, 2]$) a tzv. rovnostářský bod R (s hodnotou souřadnic po zaokrouhlení $[3,51; 3,51; 3,51]$), který popisuje takové rozdělení výher, kdy všichni tři hráči dostanou stejnou odměnu.

Model elementárního redistribučního systému umožňuje popis různých typů vyjednávání a výsledky tohoto vyjednávání zobrazovat jako vyjednávací trajektorie na redistribuční ploše.

Z pohledu teorie her je výše popsaná situace v redistribučním systému hrou s nekonstantním součtem tří inteligentních hráčů, ve kterých lze vytvářet koalice a je možné hrát hru opakovaně.

4 Použití programu Sage v analýze redistribučního systému

Pro zobrazení redistribuční plochy, její analýzu a výpočet významných bodů redistribučního systému byl použit matematický program Sage. Sage znázorňuje obrázky programem Jmol [3], který je v programovacím jazyce Java, a díky tomu lze pomocí myši obrázky různě otáčet a přibližovat, a tím lépe analyzovat celý objekt.

4.1 Znázornění elementární redistribuční plochy

Pro znázornění plochy v trojrozměrném prostoru se obvykle používá funkce `plot3d`. To lze využít pouze tehdy, jestliže je hodnota z funkcí dvou proměnných x a y , tedy $z = f(x, y)$. Tuto podmínku bohužel nespĺňuje rovnice elementárního redistribučního systému, protože jde o funkci tří proměnných x, y, z . V důsledku toho se pro znázornění redistribuční plochy musí použít funkce `implicit_plot3d`, která zobrazuje rovnici ve tvaru $f(x, y, z) = 0$ [5]. Povinnými vstupními argumenty příkazu je samotná znázorňovaná funkce a také rozměr os x, y, z . Rozměr os lze zapsat buď trojicí údajů (x , x_{\min} , x_{\max}) nebo postačuje uvedení minimální a maximální hodnoty na konkrétní ose např. (x_{\min} , x_{\max}). Dále je možné doplnit další požadavky na podobu zobrazovaného objektu, a to např. barvu, zobrazení os, velikost a průhlednost objektu.

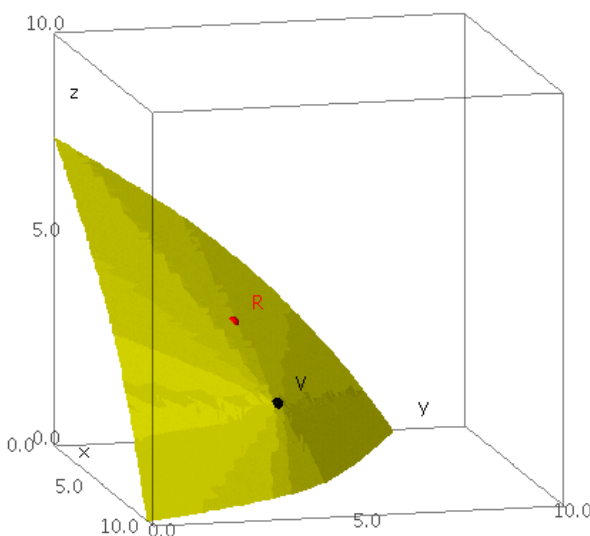
Zobrazení popisek os v 3D prostoru ovšem není úplně nejsnazší. První možností, která je uvedena i v následujícím příkladě je, že se popisky tří os znázorní jako textová pole, ovšem každé toto textové pole se musí přesně umístit v souřadnicovém systému. Tento způsob není příliš praktický, protože při jakékoliv změně v obrázku se mohou popisky skrýt a znova se musí hledat souřadnice, ve kterých je popisek osy vidět. Druhou možností je po vygenerování obrázku použít pravé tlačítko myši a zvolit `Style/Axes`. Tím se uprostřed náčrtu barevně zobrazí tři pomocné osy s popiskem. Přestože lze měnit barvu tohoto pomocného útvaru, tak velmi snižuje přehlednost celého obrázku.

Pro zobrazení popisek bodů v souřadnicovém systému jsem také používala popisek typu textové pole, pro jehož umístění se musí rovněž určit souřadnice. Na Obr. 5 jsou kromě samotné redistribuční plochy a popisek os znázorněny i dva významné body. Bod označený písmenem V je bod rozdělení výplat podle výkonnosti, kdy každý účastník systému získá přesně tolik, kolik odpovídá jeho skutečněmu výkonu. V tomto bodě je součet výplat

```
# Zakladni redistribucni plocha, rucni popisky os, bod V (deleni vykonove), bod R (deleni rovnostarske)
x, y, z = var ('x, y, z')
f = x + y + z - 12 + 0.5 * sqrt((x-6)^2 + (y-4)^2 + (z-2)^2)
osax = text3d('x', (1, 0.5, 0)); osay = text3d('y', (0, 9, 0.5)); osaz = text3d('z', (0, 0.5, 8.5))
V = (6, 4, 2); R = (3.51, 3.51, 3.51)
implicit_plot3d (f, (0,10), (0,10), (0,10), color = 'yellow') + osax + osay + osaz + point3d (V, size = 8, color =
'black') + point3d (R, size = 8, color = 'red') + text3d ('V', (6.5, 4.5, 2.5), color = 'black') + text3d ('R', (4,
4, 4), color = 'red')
```

Obrázek 4: Příkaz pro znázornění redistribuční plochy, popisků os, bodů V a R

všech hráčů největší. Čím dále od tohoto bodu, tím více hodnota součtu výplat klesá. Bod označený písmenem R je bod rovnostářského rozdělení výplat, kdy všichni účastníci daného systému získají stejnou výplatu. Těmito dvěma body musí procházet každá redistribuční plocha.



Obrázek 5: Redistribuční plocha

4.2 Výpočet důležitých bodů redistribuční plochy

Elementární rovnice redistribuční plochy obsahuje odmocninu, což představuje významný problém při matematických výpočtech v aplikaci Sage. Dosud jsem neobjevila jinou možnost řešení, než si rovnici pro konkrétní příklad umocnit a až potom zadávat upravený tvar rovnice. Tento způsob není vůbec praktický a může s sebou nést i chyby vzniklé v průběhu ručního výpočtu, včetně případné chyby zaokrouhlovací.

Celkem jsem analyzovala a hledala souřadnice následujících bodů redistribuční plochy:

- bodu R , který popisuje rovnostářské rozdělení výplat v systému,

- tři průsečíků redistribuční plochy s osami souřadnic x, y, z (body P_x, P_y, P_z),
- tři důležitých bodů ležících v jednotlivých souřadnicových plochách (body X, Y, Z).

Analýza bodu R , který popisuje rovnostářské rozdělení výplat v systému

Tento bod popisuje takové rozdělení, kde všichni tři účastníci dostanou stejnou výplatu ($x = y = z$). Redistribuční rovnice se proto vyjádří jako funkce jediné proměnné x . Na Obr. 6 je vidět, že si Sage neporadil s vyřešením rovnice obsahující odmocninu. Tvar rovnice pouze částečně upravil, ale ponechal ve výrazu druhou odmocninu a také neznámou proměnnou x . Následně jsem již zadávala umocněný a upravený formát rovnice. Umocněním se bohužel vždy získá druhý kořen rovnice, který z hlediska jeho interpretace ve vztahu ke zkoumanému objektu není správný. Správným kořenem rovnice je přibližná hodnota $x = 3,51$. Souřadnice bodu R jsou tak $[3,51; 3,51; 3,51]$. Druhý kořen rovnice není správný, protože suma souřadnic by převyšovala hodnotu 12, která je maximální možná, a kořen by nevyhovoval původní redistribuční rovnici.

```
# SOURADNICE ROVNOSTARSKÉHO BODU (x = y = z)
# Nevyřešení rovnice s odmocninou
x = var('x')
g = -3*x + 12 - 0.5*sqrt((x - 6)**2 + (x - 4)**2 + (x - 2)**2)
solve(g,x)

[x == -1/6*sqrt(3*x^2 - 24*x + 56) + 4]

# Uprava původní rovnice
x = var('x')
g = 8.25*x^2 - 66*x + 130
s = solve(g, x, solution_dict = True)
x1, x2 = float(s[0][x]), float(s[1][x])
print x1, x2

3.50763403608 4.49236596392

# Ověření vypočtu dosazením do původní rovnice
x = var('x')
g = -3*x + 12 - 0.5*sqrt((x - 6)**2 + (x - 4)**2 + (x - 2)**2)
g.substitute(x = x1); g.substitute(x = x2)

-8.88178419700125e-16
-2.95419578350399
```

Obrázek 6: Příkaz pro zjištění souřadnic rovnostářského bodu

Analýza průsečíků redistribuční plochy s osami souřadnic

Pro průsečík redistribuční plochy s osou z (bod P_z) platí, že ostatní souřadnice tohoto průsečíku jsou nulové ($x = y = 0$). Tyto hodnoty se dosadí do rovnice redistribuční plochy a opět tím vznikne rovnice o jedné neznámé. Rovnice se musela nejprve umocnit a až potom vyřešit programem Sage. Umocněním rovnice vzniknou opět dva kořeny řešení, ale jeden z kořenů nevyhovuje původní elementární redistribuční rovnici, což lze v programu Sage ověřit. Stejný postup výpočtu byl použit i pro zbylé dva průsečíky P_x a P_y .

```
# PRUSECÍK PLOCHY S OSOU z (x = y = 0)
# Sage nevyřeší rovnici s odmocninou
z = var('z')
g = -z + 12 - 0.5*sqrt((6)**2 + (4)**2 + (z- 2)**2)
solve(g,z)

[z == -1/2*sqrt(z^2 - 4*z + 56) + 12]

# Uprava puvodni rovnice
z = var('z')
g = 0.75*z^2 - 23*z + 130
s = solve(g, z, solution_dict = True)
Pz1, Pz2 = float(s[0][z]), float(s[1][z])
print Pz1, Pz2

7.47344925163 23.193217415

# Overeni
z = var('z')
g = -z + 12 - 0.5*sqrt((6)**2 + (4)**2 + (z- 2)**2)
g.substitute(z = Pz1); g.substitute(z = Pz2)

0.000000000000000
-22.3864348300688

Pz = (0, 0, Pz1)
```

Obrázek 7: Příkaz pro výpočet průsečíku redistribuční plochy s osou z

Analýza důležitých bodů ležících v jednotlivých souřadnicových plochách

Jde o velmi významnou skupinu bodů, které jsou důležité při analýze vyjednávání v redistribučních systémech. Na Obr. 8 je uveden příkaz pro výpočet souřadnic bodu Z , který leží v rovině xy , a tak je jeho z -ová souřadnice nulová.

Proměnná y se vyjádří pomocí proměnné x (např. z rovnice $y = ax$ a dosazením souřadnic bodu $V = [6, 4, 2]$; $4 = 6a$; $a = 2/3$; $y = 2/3x$).

Elementární redistribuční rovnice se vyjádří jako rovnice jediné proměnné x . Vypočítané hodnoty x_1 , x_2 a y_1 , y_2 se dosadí do původní elementární redistribuční rovnice, a tím se zjistí správné kořeny dané rovnice, které jsou potom hledanými souřadnicemi bodu Z . Všechny tyto vypočtené důležité body lze znázornit na redistribuční ploše (příkaz pro

```
# Vypocet souradnic bodu Z v plose XY, z = 0, y = 2/3x
x = var('x')
g = (5/3*x - 12)**2 - 0.25*((x-6)**2+(2/3*x-4)**2+4)
s = solve(g, x, solution_dict = True)
x1, x2 = float(s[0][x]), float(s[1][x])
print x1, x2

6.5662188881 8.19240180156

y1 = 2/3 * x1; y2 = 2/3 * x2; print y1, y2

4.37747925873 5.46160120104

# Zpetne dosazeni x1 a y1 do rovnice redistribucni plochy (lepsi vysledek)
r = 12 - x1 - y1 - 0.5*sqrt((x1 - 6)**2 + (y1 - 4)**2 + (0 - 2)**2); print r

2.44249065417534e-15

# Zpetne dosazeni x1 a y1 do rovnice redistribucni plochy (horsí vysledek)
r = 12 - x2 - y2 - 0.5*sqrt((x2 - 6)**2 + (y2 - 4)**2 + (0 - 2)**2); print r

-3.30800600518427

# SOURADNICE bodu Z v plose XY (z = 0)
Z = (x1, y1, 0); print Z

(6.5662188880998889, 4.3774792587332589, 0)
```

Obrázek 8: Příkaz pro výpočet souřadnic bodu Z , ležícího v rovině xy

zobrazení Obr. 9, znázornění Obr. 10). Průřezíky s osami souřadnic jsou zobrazeny modrou barvou, důležité body v jednotlivých souřadnicových plochách zelenou barvou.

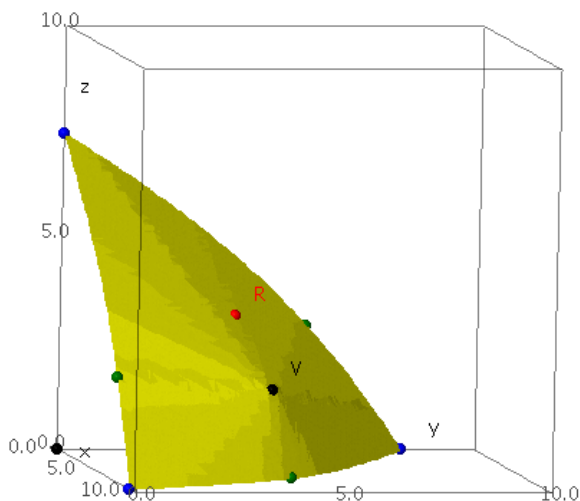
```
print Px, Py, Pz
(9.2390477143047676, 0, 0) (0, 8.2030935234448936, 0) (0, 0,
7.4734492516322693)

print X, Y, Z
(0, 5.8807373824995413, 2.9403686912497706) (7.4010311744816315, 0,
2.467010391493877) (6.5662188880998889, 4.3774792587332589, 0)

# Znázornění důležitých bodů, průřezů s osami (Px, Py, Pz), počátku P, bodu V a R, průřezů v souřadnicových
plochách (X, Y, Z)
x, y, z = var ('x', 'y', 'z')
f = x + y + z - 12 + 0.5 * sqrt((x-6)^2 + (y-4)^2 + (z-2)^2)
osax = text3d('x', (1, 0.5, 0)); osay = text3d('y', (0, 9, 0.5)); osaz = text3d('z', (0, 0.5, 8.5))
P = (0, 0, 0); V = (6, 4, 2); R = (3.51, 3.51, 3.51)

implicit plot3d (f, (0,10), (0,10), (0,10), color = 'yellow') + osax + osay + osaz + point3d (V, size = 8, color =
'black') + text3d ('V', (6.5, 4.5, 2.5), color = 'black') + point3d (R, size = 8, color = 'red') + text3d ('R', (4,
4, 4), color = 'red') + point3d (P, size = 8, color = 'black') + point3d (Px, size = 8, color = "blue") + point3d
(Py, size = 8, color = "blue") + point3d (Pz, size = 8, color = "blue") + point3d (X, size = 8, color = "green") +
point3d (Y, size = 8, color = "green") + point3d (Z, size = 8, color = "green")
```

Obrázek 9: Příkaz pro znázornění redistribuční plochy a důležitých bodů



Obrázek 10: Redistribuční plocha a důležité body

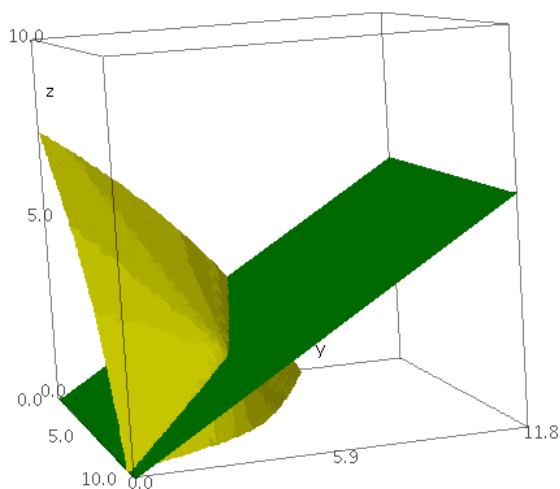
4.3 Znázornění pomocných vyjednávacích rovin

Při analýze redistribučních systémů je velmi důležité sledovat různé typy vyjednávání jednotlivých hráčů. Na redistribuční ploše lze jednotlivé vyjednávací trajektorie znázornit. Základní jsou tři vyjednávací trajektorie při dohodě dvou hráčů o rozdělení svých odměn podle jejich výkonu. V této situaci si dva hráči dělí výplatu v poměru ke svým výkonům


```
# Znazornění jedné ze tří pomocných rovin
x, y, z = var('x, y, z')
f = x + y + z - 12 + 0.5 * sqrt((x-6)^2 + (y-4)^2 + (z-2)^2)
X1 = (0, 2*5.88073738249954, 2*2.94036869124977)
X2 = (10, 2*5.88073738249954, 2*2.94036869124977)
P = (0, 0, 0)
Pxx = (10, 0, 0)
osax = text3d('x', (1, 0.5, 0)); osay = text3d('y', (0, 9, 0.5)); osaz = text3d('z', (0, 0.5, 8.5))
implicit_plot3d(f, (0,10), (0,10), (0,10), color='yellow') + polygon3d([X2, X1, P, Pxx], color="green") + osax
+ osay + osaz
```

Obrázek 11: Příkaz pro znázornění redistribuční plochy a jedné z pomocných rovin

a třetímu zůstane jen tolik, kolik mu ostatní dva ponechají. Trajektorie tohoto vyjednávání se protínají v jednom bodě, a to v bodě V se souřadnicemi $[6, 4, 2]$. Trajektorie začínají v bodech X, Y, Z , které byly vypočteny, a končí v průsečících redistribuční plochy s osami souřadnic, tedy v bodech P_x, P_y, P_z . Pomocná rovina byla znázorněna s využitím funkce



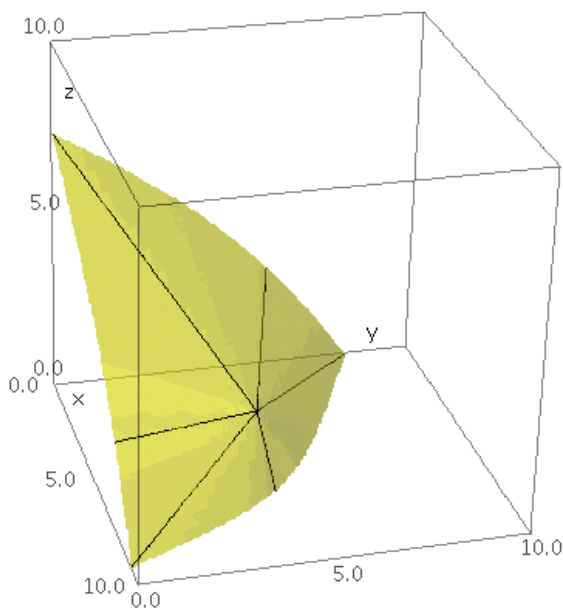
Obrázek 12: Redistribuční plocha a jedna z pomocných rovin

`polygon3d`. Základním argumentem této funkce jsou body, které se mají spojit do jednoho útvaru. Při zpracování této části jsem zjistila, že podoba výsledného útvaru záleží na pořadí bodů v zadání příkazu, což může být významná komplikace při řešení matematických úloh. Vyjednávací trajektorii tvoří společné body této pomocné roviny a redistribuční plochy.

Všechny tři pomocné roviny rozdělí redistribuční plochu na šest dílčích oblastí. Každá pomocná rovina prochází bodem V , určujícím rozdělení podle výkonu, a počátkem souřadnicového systému. Vyjednávací trajektorie popisující situaci dělení podle výkonu jsou celkem tři. Každou z trajektorií lze získat jako průsečnici redistribuční plochy a pomocné roviny tak, jak je ukázáno na Obr. 14.

```
# Vyjednavací trajektorie
x, y, z = var("x y z")
f = x + y + z - 12 + 0.5 * sqrt((x - 6)**2 + (y - 4)**2 + (z - 2)**2)
osax = text3d('x', (1, 0.5, 0)); osay = text3d('y', (0, 9, 0.5)); osaz = text3d('z', (0, 0.5, 8.5))
p = implicit_plot3d(f, (0,10), (0,10), (0,10), color = 'yellow', opacity = 0.6)
l11 = line3d([(6.56621888809989, 4.37747925873326, 0), (6,4,2)], color = "black")
l12 = line3d([(6,4,2), (0,0,7.47)], color = "black")
l21 = line3d([(7.40103117448163, 0, 2.46701039149388), (6,4,2)], color = "black")
l22 = line3d([(6,4,2), (0,8.2,0)], color = "black")
l31 = line3d([(0, 5.88073738249954, 2.94036869124977), (6,4,2)], color = "black")
l32 = line3d([(6,4,2), (9.24,0,0)], color = "black")
show(p + l11 + l12 + l21 + l22 + l31 + l32 + osax + osay + osaz)
```

Obrázek 13: Příkaz pro zobrazení vyjednávacích trajektorií



Obrázek 14: Vyjednávací trajektorie na redistribuční ploše

Závěr

Matematický program Sage se ukázal být velmi vhodným softwarem pro grafické znázornění redistribučních systémů. Práce v grafickém rozhraní je uživatelsky velmi příjemná. Syntaxe příkazů je logická. Realizace některých matematických operací a popisků grafických objektů nebyla uživatelsky úplně jednoduchá, ale to může být i tím, že jsem doposud možná neobjevila všechny dostupné funkce a možnosti programu. Rovněž se na vývoji programu Sage neustále pracuje, a tak je pravděpodobné, že se systém stane ještě dokonalejším.

V případě redistribučních systémů je tato práce pouhým úvodem k jejich rozboru. S využitím programu Sage je vhodné zkoumat různé typy vyjednávacích trajektorií, vliv tvorby koalic mezi účastníky na možnost ovlivnění redistribuce a hledání rovnovážných situací.

Literatura

- [1] BUDINSKÝ, P. – VALENČÍK, R.: Teorie redistribučních systémů. In: Politická ekonomie, roč. 13, 2009, č. 5, s. 644 – 659
- [2] HAUPT, M.: Systém počítačové algebry Sage. Bakalářská práce. Brno, 2009. Dostupné na internetu [cit. 2010-06-08]: <http://www.math.muni.cz/~{ }plch/diplomky/Sage.pdf>
- [3] Jmol. Dostupné na internetu [cit. 2010-06-10]: <http://jmol.sourceforge.net/>
- [4] PYTHON – programovací jazyk. Dostupné na internetu [cit. 2010-06-08]: http://sk.wikipedia.org/wiki/Python_%28programovac%C3%AD_jazyk%29
- [5] SAGE. Oficiální stránka. Dostupné na internetu [cit. 2010-06-08]: <http://sagemath.org/>
- [6] SAGE Reference Manual, 3D Graphics. Dostupné na internetu [cit. 2010-06-08]: <http://www.sagemath.org/doc/reference/plot3d.html>

Kontaktní adresa

Karolína MUŽÍKOVÁ (Ing.),
Katedra manažérských teorií FRI ŽU v Žiline,
Univerzitná 8215/1, 010 26 Žilina,
Karolina.Muzikova@fri.uniza.sk

Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach

1.–4. júla 2010, Žilina, Slovensko

Organizátori: Miloš Šrámek, Spoločnosť pre otvorené informačné technológie
Tatiana Šrámková, Katedra fyziky, FEI STU Bratislava
Michal Kaukič, Aleš Kozubík, Tomáš Majer, Žilinská univerzita
Lýdia Gábrisová, Ľubica Micháľková, Žilinská univerzita
Juraj Bednár, Digmia, Slovensko
Miloslav Ofúkaný, GeoCommunity, Slovensko
Peter Mráz, Kremnica
Slavko Fedorik, SOŠ elektrotechnická, Poprad
Peter Štrba, Spojená škola/Gymnázium M. Galandu, Turčianske Teplice
Ladislav Ševčovič, FEI, Technická univerzita v Košiciach

Editori: Michal Kaukič
Miloš Šrámek
Slavko Fedorik
Ladislav Ševčovič

Recenzenti: Mgr. Juraj Bednár
Mgr. Rudolf Blaško, PhD.
RNDr. Ján Buša, CSc.
Ing. Slavko Fedorik
Ing. Karol Grondžák, PhD.
Mgr. Michal Kaukič, CSc.
Ing. Tomáš Kliment
RNDr. Aleš Kozubík, PhD.
Mgr. Juraj Michálek
doc. RNDr. Štefan Peško, CSc.
Ing. Pavel Stříž, PhD.
RNDr. Ladislav Ševčovič
Ing. Michal Žarnay, PhD.

Vydavateľ: Spoločnosť pre otvorené informačné technológie – SOIT, Bratislava

ISBN 978-80-970457-0-8

Sadzba programom pdfT_EX Ladislav Ševčovič

Copyright © 2010 autori príspevkov. Príspevky neprešli redakčnou ani jazykovou úpravou.

Ktokoľvek má dovolenie vyhotoviť alebo distribuovať doslovný opis tohoto dokumentu alebo jeho časti akýmkoľvek médiom za predpokladu, že bude zachované oznámenie o copyrighte a o tom, že distribútor príjemcovi poskytuje povolenie na ďalšie šírenie, a to v rovnakej podobe, akú má toto oznámenie.