

Úloha VI.S ... Matice a populace

10 bodů; (chybí statistiky)

- a) Na základě Lotkova-Volterrova modelu simulujte vývoj populace predátora a kořisti (např. sluněčka sedmitečného a mšice makové) pro následující hodnoty parametrů: $r_m = 0,8$, $D_m = 1,0$, $r_s = 0,75$, $D_s = 1,5$. Počáteční populace volte po dvojicích jako $m = 0,5$ a $s = 2,0$; $m = 1,5$ a $s = 0,5$; $m = 1,95$ a $s = 0,75$. Výsledek zanešte do grafu závislosti populace predátora na populaci kořisti. Výsledek diskutujte.
 Bonus Naleznete tvar křivek v grafu pomocí analytických metod (integrací diferenciální rovnice).
- b) Použitím kompetitivního Lotkova-Volterrova modelu simulujte vývoj dvou soupeřících populací s omezenou populační kapacitou (např. káně lesní a poštolka obecná) pro tyto hodnoty parametrů: $r_k = 0,8$, $I_{kp} = 0,2$, $k_k = 2,0$, $r_p = 0,6$, $I_{pk} = 0,3$, $k_p = 1,0$. Počáteční populace volte jako $k = 0,01$, $p = 1,0$. Poté změňte interakční koeficienty na $I_{kp} = 1,5$ a $I_{pk} = 0,6$, zbytek ponechejte. Výsledky zanešte do jednoho grafu závislosti velikosti populací na čase, diskutujte.
- c) Ověřte důležitost pivotizace. Vyřešte soustavu

$$\begin{pmatrix} 10^{-20} & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

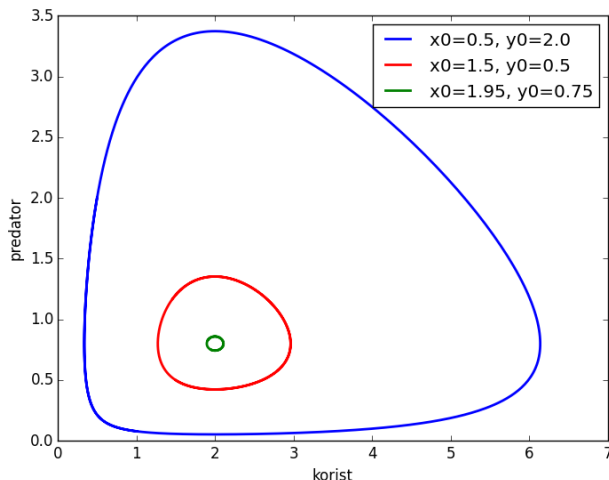
nejprve přesně (na papíře), poté s využitím LU dekompozice s (částečnou) pivotizací (využijte nějakou knihovní funkci, např. `scipy.linalg.lu()`), a nakonec pomocí LU dekompozice bez pivotizace (to si budete muset sami naprogramovat). Porovnejte výsledky \mathbf{x} z jednotlivých metod a výsledky zpětného vynásobení matic $L^{-1} \cdot U$ (resp. $P \cdot L^{-1} \cdot U$ v případě s pivotizací).

- d) Mějme nekonečný deskový kondenzátor se vzdáleností desek $L = 10$ cm a napětím mezi deskami $U = 5$ V. Do kondenzátoru vložíme uzemněnou elektrodu ve tvaru nekonečně dlouhé hranoly s čtvercovou podstavou o hraně $a = 2$ cm, jejíž střed leží $l = 6,5$ cm od uzemněné desky původního kondenzátoru (tak, že leží mezi deskami). Hranol je orientován tak, že jedna z jeho kratších hran je kolmá k deskám kondenzátoru. Naleznete průběh elektrického potenciálu v kondenzátoru. Protože je problém symetrický vůči posunu v ose rovnoběžné s nekonečnou hranou hranoly, stačí jej řešit v řezu kolmém k této ose, jde tedy o 2D problém. V této rovině pak získaný průběh potenciálu také vykreslete. K řešení můžete využít program přiložený k zadání.

Bonus Vypočtete a vykreslete také průběh velikosti intenzity el. pole \mathbf{E} .

Mirek a Lukáš naplňují matice attoliškami.

- a) S využitím kódu z šestého dílu seriálu jsme spočetli časový vývoj populací kořisti a predátora pro zadané parametry a počáteční hodnoty. Populaci predátora jsme pak vykreslili jako proměnnou závislou na populaci kořisti. Výsledný graf je na obrázku 1. Jelikož jsou řešení Lotkova-Volterrova modelu predátor-kořist periodická, dostáváme v rovině xy uzavřené křivky. Vidíme, že případ s vysokou počáteční populací predátora předvídá, že v důsledku nedostatku kořisti klesne během vývoje populace predátora do velmi nízkých hodnot. Také si všimneme nepravidelného tvaru křivky. Pro druhou volbu počátečních hodnot je křivka již pravidelnější a pro třetí volbu je téměř eliptická. Dalším postřehem je,



Obr. 1: Vývoj populace predátora a kořisti pro různé počáteční podmínky.

že se křivky nekříží – kdyby tomu tak nebylo, neměla by soustava diferenciálních rovnic jednoznačné řešení.

Protože se křivky nekříží a jsou uzavřené, postupnou změnou počátečních populací můžeme dokonvergovat do jednoho bodu. Jedná se o stacionární bod

$$(x, y) = (D_s/r_s, r_m/D_m) = (2, 0; 0, 8).$$

Pokud tedy nastartujeme simulaci poblíž tohoto bodu, dostaneme téměř eliptické křivky, neboť pro malé změny populací můžeme Lotkovy–Volterrovy rovnice nahradit harmonickým oscilátorem, jehož řešením ve fázovém prostoru jsou právě elipsy. Časový vývoj probíhá proti směru hodinových ručiček.

Řešení bonusové úlohy začíná vydělením diferenciální rovnice pro predátora rovnicí pro kořist. Diferenciály času se zkrátí¹ a dostaneme jednu rovnici

$$\frac{dy}{dx} = \frac{y(r_y x - D_y)}{x(r_x - D_x y)}.$$

Na pravé straně jsme rovnou provedli vytknutí, aby bylo zřejmé, že rovnici lze řešit separací proměnných. Integrujeme²

$$\int dy \frac{r_x - D_x y}{y} = \int dx \frac{r_y x - D_y}{x}.$$

¹Možná už jste někde zaslechli, že za výroky typu „ dx se nám vykrátí“ se vyhazuje z Matfyzu, zvlášť pokud studujete nějaký matematický obor. Je pravda, že diferenciály jakožto samostatné matematické objekty dělit nelze (operace není definována), ale v Leibnizově notaci můžeme dát dělení diferenciálů dobrý význam na základě řetězového pravidla $\frac{dy(x(t))}{dt} = \frac{dy}{dx} \frac{dx}{dt}$.

²„A tady už jako můžeme pracovat s diferenciály jako se zlomkem?“ Ne. Jedná se opět o řetězové pravidlo. Jak zde bylo použito vám necháme na rozmyšlení.

Obě strany mají podobný tvar, podívejme se proto jen nalevo. Integrál z D_x je triviálně yD_x , integrace r_x/y vede na $r_x \ln y$. Nezapomeneme na konstantu a dostaneme

$$r_x \ln y - D_x y = r_y x - D_y \ln x + C.$$

V tomto tvaru již rovnici necháme, neboť nemá analytické řešení. Kdybychom bývali chtěli, mohli jsme numerické řešení Lotkových-Volterrových rovnic vynechat a použít rovnou toto exaktní řešení, pouze bychom museli určit konstantu C z počátečních podmínek.

Zde i v seriálu jsme zmínili, že poblíž stacionárního bodu se řešení chová stejně jako harmonický oscilátor. Padlo u toho slůvko „linearizace“. Linearizací se myslí rozvoj řešení okolo stacionárního bodu do prvního (lineárního) řádu. Abyste si nemysleli, že se jedná o nějakou magii, ukážeme si zde odvození. Píšeme $x(t) = x_0 + x_1(t)$, kde x_0 je souřadnice stacionárního bodu, na čase samozřejmě nezávislá, a $x_1(t)$ je na čase závislá odchylka o stacionární polohy. Podobně pro $y(t)$. Po dosazení do Lotkových-Volterrových rovnic máme

$$\dot{x}_1 = r_x(x_0 + x_1) - D_x(x_0 + x_1)(y_0 + y_1).$$

Použili jsme zde značení časové derivace tečkou. Jelikož odchylky od stacionárního bodu jsou malé, můžeme zanedbat člen $x_1 y_1$ a po dosazení známých hodnot x_0, y_0 nám zbude

$$\begin{aligned} \dot{x}_1 &\approx r_x x_1 - D_x x_0 y_1 - D_x x_1 y_0 \\ &= r_x x_1 - D_x \frac{D_y}{r_y} y_1 - D_x x_1 \frac{r_x}{D_x} \\ &= -\frac{D_x D_y}{r_y} y_1, \\ \dot{y}_1 &\approx \frac{r_x r_y}{D_x} x_1. \end{aligned}$$

První rovnici zderivujeme podle času a napravo dosadíme z druhé rovnice. Podobnou operaci uděláme s druhou rovnicí a dostaneme rovnice harmonického oscilátoru

$$\begin{aligned} \ddot{x}_1 + r_x D_y x_1 &= 0, \\ \ddot{y}_1 + r_x D_y y_1 &= 0. \end{aligned}$$

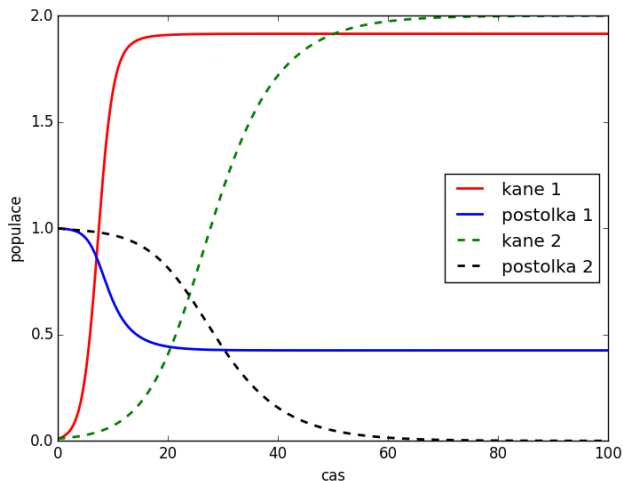
Všimněte si, že frekvence $\sqrt{r_x D_y}$ závisí pouze na parametrech, které nevystupují v interakčních členech.

- b) Kompetitivní model vyžaduje pouze přidání jednoho koeficientu, úpravu předpisů diferenciálních rovnic a výstupu z funkce, jinak se simulační kód nijak neliší od předchozího případu. Pro dané koeficienty jsme dostali graf na obrázku 2.

Pro malé hodnoty interakčních parametrů se velice rychle ustanoví rovnováha. Přesné velikosti populací, ke kterým řešení konverguje, získáme vypočtením polohy stacionárního bodu

$$(x, y) = \left(\frac{k_x - I_{xy} k_y}{1 - I_{xy} I_{yx}}, \frac{k_y - I_{yx} k_x}{1 - I_{xy} I_{yx}} \right) \doteq (1,915; 0,426).$$

Pro druhou volbu parametrů dostaneme stacionární bod $(x, y) = (5,0; -2,0)$. Vidíme, že zde již nemůže dojít ke konvergenci ke stacionárnímu bodu, protože v něm vystupuje záporná hodnota. Druhá populace proto poměrně rychle (exponenciálně) konverguje k nule a první populace ke svojí nosné kapacitě. Kdybychom volili v druhém případě $I_{pk} = 0,5$ místo 0,6, dostali bychom speciálně stacionární hodnotu $y = 0$ a druhá populace by se v limitě $t \rightarrow \infty$ chovala jako $1/t$, což je pomalejší pokles než ten, který pozorujeme s původními parametry. Za povšimnutí také stojí, že pokud jsou organismy schopny koexistovat, jejich populace může být pro určité hodnoty parametrů (např. i v této úloze) v součtu větší, než když jedna z populací vyhyne.



Obr. 2: Časový vývoj dvou soupeřících populací s různými interakčními parametry.

- c) Označme matici soustavy se zadání jako A a vektor pravých stran \mathbf{b} . Nejprve provedeme LU dekompozici na papíře, budeme přitom postupovat dle algoritmu představeného v seriálu. Protože máme matici 2×2 , bude $L = L_1$. Jediný netriviální prvek je $l_{21} = a_{21}/a_{11} = 10^{20}$. Matice L tedy je

$$L = \begin{pmatrix} 1 & 0 \\ -10^{20} & 1 \end{pmatrix}$$

a hledaná matice L^{-1} je

$$L^{-1} = \begin{pmatrix} 1 & 0 \\ 10^{20} & 1 \end{pmatrix}.$$

Matici U bychom dokázali spočítat součinem $U = LA$, maticové násobení je ale náročná operace (alespoň pro matice vyššího řádu, ale i pro 2×2 matice bychom potřebovali 8 násobení), k problému tedy přistoupíme jinak. Z popisu algoritmu je jasné, že první řádek bude mít matice U shodný s maticí A . Protože naším cílem bylo vynulovat první sloupec pod diagonálou, bude $u_{21} = 0$ (matice značíme velkým písmem, jejich prvky malým). Zbývající prvek vypočteme pravidlem pro maticové násobení, protože má ale každá z matic L_k na každém řádku jen dva nenulové prvky, z nichž jeden je jednička na diagonále, redukuje se maticové násobení na $a_{ij}^{(k)} = a_{ij}^{(k-1)} - l_{ik}a_{kj}^{(k-1)} \forall i, j > k$. V našem případě tedy platí $u_{22} = a_{22} - l_{21}a_{12}$ a matice U je

$$U = \begin{pmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{pmatrix}.$$

Nyní vypočteme vektor \mathbf{x} , tedy vyřešíme rovnice $L^{-1}\mathbf{y} = \mathbf{b}$ a $U\mathbf{x} = \mathbf{y}$. Protože matice jsou v trojúhelníkovém tvaru, můžeme použít metodu dopředné a zpětné substituce, jak jsme si ji představili v seriálu. Platí tedy rovnice

$$\begin{aligned}y_1 &= \frac{b_1}{l_{11}} = 1, \\y_2 &= \frac{b_2 - y_1 l_{21}}{l_{22}} = -10^{20}, \\x_2 &= \frac{y_2}{u_{22}} = \frac{10^{20}}{10^{20} - 1}, \\x_1 &= \frac{y_1 - x_2 u_{12}}{u_{11}} = \frac{10^{20}}{1 - 10^{20}},\end{aligned}$$

výsledek tedy je

$$\mathbf{x} = \begin{pmatrix} \frac{10^{20}}{1 - 10^{20}} \\ \frac{10^{20}}{10^{20} - 1} \end{pmatrix} \approx \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

Lehce si pak také ověříme, že $L^{-1}U = A$.

Nyní provedeme LU rozklad s částečnou pivotizací. Použijeme k tomu metodu `scipy.linalg.lu()`. Z její dokumentace zjistíme, že má jeden povinný a tři nepovinné parametry. Povinným parametrem je matice A , kterou chceme rozložit, nepovinné parametry měnit nebudeme. Dále se z dokumentace dozvíme, že metoda provede rozklad $A = PLU$, kde P je permutační matice (reprezentace pivotizace), L je dolní trojúhelníková matice (to, co zde značíme L^{-1}) a U horní trojúhelníková matice. Tyto tři matice metoda vrací v tomto pořadí. Při použití LU rozkladu s částečnou pivotizací dostaneme stejné³ (tedy správné) výsledky, jen kvůli pivotizaci nyní máme rozklad $PL^{-1}U\mathbf{x} = \mathbf{b}$, při hledání \mathbf{x} tedy musíme nejprve vypočítat $\tilde{\mathbf{b}} = P^{-1}\mathbf{b} = P^T\mathbf{b}$, kde P^T je transponovaná permutační matice. Tím soustavu převedeme do tvaru $L^{-1}U\mathbf{x} = \tilde{\mathbf{b}}$, který už umíme řešit.

Nakonec provedeme LU dekompozici bez pivotizace. V podstatě jde o stejný postup, jaký jsme před chvílí prováděli na papíře, tentokrát jej ale chceme počítat se strojovou přesností a ne přesně. Jeho provedením zjistíme, že matice L je stejná, jako v přesném výpočtu, protože při jejím výpočtu pouze dělíme, což je z hlediska numerických chyb bezpečná operace. Pokud bychom ale počítali větší matici než 2×2 , pak by výpočet L_k obsahoval i odčítání, chyby by se tedy objevily i zde. Matice U ale již nyní vypadá

$$U = \begin{pmatrix} 10^{-20} & 1 \\ 0 & -10^{20} \end{pmatrix}.$$

Na první pohled jde o velmi malou relativní chybu jen v jednom z prvků matice, jak ale uvidíme, tato chyba způsobí velkou chybu výsledku. Zpětným vynásobením totiž dostaneme

$$L^{-1}U = \begin{pmatrix} 10^{-20} & 1 \\ 1 & 0 \end{pmatrix},$$

což není původní matice A . Ještě výraznější chyba je v \mathbf{x} , který nyní je

$$\mathbf{x} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

³Matice L^{-1} a U jsou kvůli permutaci samozřejmě zcela jiné.

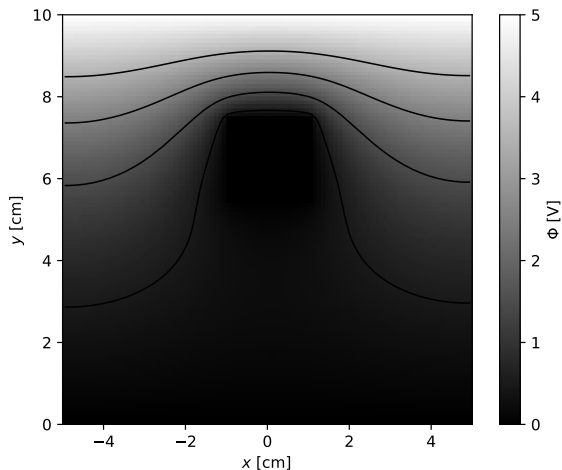
Jasně tedy vidíme, že LU dekompozice bez pivotizace je numericky nestabilní.

- d) Skript přiložený k zadání řeší velmi podobnou úlohu, pouze místo elektrody ve tvaru hranolu má elektrodu ve tvaru nekonečně vysokého válce, skript tedy potřebujeme lehce modifikovat. Nejprve si všimneme, že potenciál na horní desce je 1 V místo 5 V, což lehce upravíme v definici pravé strany zobecněné Dirichletovy podmínky v oblasti horní desky, tedy na řádku 39. Dále potřebujeme upravit rozměry simulované oblasti. Toho lze docílit úpravou hodnot h , $X0$ a $Y0$, v našem případě je ale jednodušší prostě v popisku grafu zaměnit metry za centimetry.⁴ Hodnota h , vyjadřující vzdálenost dvou sousedních bodů mřížky, nyní nebude v metrech, ale v centimetrech. Nakonec nahradíme válec hranolem, tedy změním oblast, kde aplikujeme Dirichletovu okrajovou podmínku. Toho docílíme změnou podmínky na řádku 44, kde výraz $(j-Ny/4)**2+(i-Nx/2)**2<=Ny/5$ (podmínka na to, že bod mřížky (i, j) leží v kruhu o poloměru $\sqrt{Ny/5}$ a středu $(Nx/2, Ny/4)$) nahradíme výrazem $abs(j-3.5/h)<=1.0/h$ and $abs(i-Nx/2)<=1.0/h$, podstava hranolu tedy bude mít stranu délky 2 cm a její střed bude vzdálen 3,5 cm od horní desky (tedy 6,5 cm od spodní, uzemněné desky). Protože je kondenzátor nekonečný, můžeme hranol v x -ovém směru umístit kamkoliv, nejlepší je ale umístit jej doprostřed simulované oblasti, jednak abychom vystihli symetrii úlohy, jednak abychom byli co nejdále od krajů, kde používáme von Neumannovu podmínku, tak, aby tato podmínka byla fyzikálně co nejvíce opodstatněná.

Zbývají dva volné parametry simulace Nx a Ny , tedy počet bodů mřížky, na které simulujeme. Pokud obě hodnoty necháme na výchozí hodnotě 100, což, jak jistě uznáte, není nijak zázračné rozlišení, bude matice soustavy řádu 10 000, protože pro každý bod mřížky budeme mít jednu rovnici. Matice tedy bude mít 100 000 000 prvků, pokud každý z nich bude typu `double`, tedy zabere 8 B paměti, celá matice zabere cca 800 MB. I s takto malým rozlišením (navíc pouze ve 2D) jsme na hranici možnosti běžného počítače (typická velikost paměti RAM je několik GB). Naštěstí je tato matice velmi řídká, čehož lze využít a ušetřit tak paměť, je pak ale třeba použít speciální (převážně iterativní) metody pro řešení dané soustavy. Pokud máme program správně nastaven, zbývá jej pouze spustit. Výslednou mapu potenciálu vidíme na obrázku 3.

Elektrickou intenzitu pak vypočítáme ze vztahu $\mathbf{E} = -\text{grad } \varphi$, musíme tedy zjistit první derivace v obou směrech. Nejjednodušší je využít dopředné diference, např. pro x -ovou složku můžeme psát $Ex=-(\text{phi}[:,1:]-\text{phi}[:, :-1])/h$, kde $\text{phi}=\mathbf{x}.\text{reshape}(Ny, Nx)$. Výraz výše říká, že od pole s uřízým prvním sloupcem odečteme pole s uřízým posledním sloupcem (prvek po prvku), v každém prvku tedy od sebe odečteme hodnoty sousedních buněk v řádku, což odpovídá dopředné diferenci. Pole Ex tedy bude mít o jeden sloupec méně, než pole phi . Obdobně provedeme dopřednou diferenci v y -ovém směru, ta zase bude mít o jeden řádek méně. Abychom získali velikost, potřebujeme (prvek po prvku) provést $E_size=np.\text{sqrt}(Ex*Ex+Ey*Ey)$. To se nám ale nepodaří, protože Ex a Ey mají jiný tvar. V obou polích tedy musíme dodatečně umazat jeden (poslední) sloupec, resp. řádek. Výsledné příkazy tedy jsou $Ex=-(\text{phi}[:, :-1]-\text{phi}[:, :-1])/h$ a $Ey=-(\text{phi}[1:, :-1]-\text{phi}[:, :-1])/h$ Další možností je nahlédnout do dokumentace NumPy a zjistit, že nabízí metodu `numpy.gradient()`, která přímo vrací gradient dodaného pole. Počítá jej přitom centrovanou diferencí tam, kde to lze a dopřednou nebo zpětnou diferencí u krajů. Výsledek je tedy přesnější, než co jsme spočítali my, navíc má stejný počet řádků a sloupců, jako původní pole phi .

⁴V případě obecné Poissonovy rovnice bychom si museli dát pozor i na správné přeskálování hodnot na pravé straně.



Obr. 3: Rozložení elektrického potenciálu v řezu kondenzátorem.

Výslednou velikost intenzity el. pole vidíme na obrázku 4. Všimněme si, že intenzita je nejvyšší u rohů elektrody, pozorujeme známý efekt „sršení“⁵. Kromě těchto oblastí je nejvyšší intenzita v oblasti, kde je mezi uzemněnou a neuzemněnou elektrodou nejmenší vzdálenost. Jde o očekávaný výsledek, neboť se zde potenciál musí změnit na krátkém úseku, jeho derivace tedy bude velká. Tyto dva jevy hezky ilustrují, proč blesk při bouřce často udeří do vysokých a (relativně) špičatých objektů, jako jsou třeba komíny a osamělé stromy.

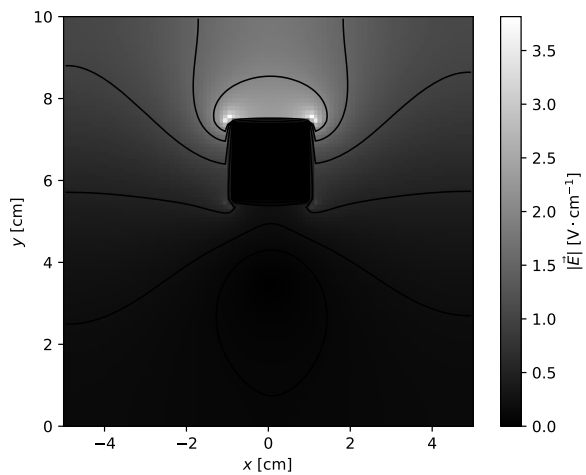
Miroslav Hanzelka
mirek@fykos.cz

Lukáš Tímko
lukast@fykos.cz

Fyzikální korespondenční seminář je organizován studenty MFF UK. Je zastřešen Oddělením pro vnější vztahy a propagaci MFF UK a podporován Ústavem teoretické fyziky MFF UK, jeho zaměstnanci a Jednotou českých matematiků a fyziků.

Toto dílo je šířeno pod licencí Creative Commons Attribution-Share Alike 3.0 Unported.
Pro zobrazení kopie této licence navštivte <http://creativecommons.org/licenses/by-sa/3.0/>.

⁵V případě přesného řešení je přímo na rohu dokonce nekonečná, nicméně my zde počítáme pouze aproximaci na mříži o konečném počtu bodů, proto dostáváme konečné hodnoty. Pokud zvýšíme počet bodů mříže, intenzita v bodech nejbližší k hranám skutečně roste do nekonečna. V reálné fyzice dokonalé hrany ovšem neexistují, ale nabitý „hrot jehly“ se skutečně využívá na dosažení vysokých intenzit.



Obr. 4: Rozložení velikosti intenzity el. pole v řezu kondenzátorem.