

# 1 Počítačové cvičení č. 1

Numerické řešení PDR

Metoda konečných diferencí

Metoda konečných prvků (matlab)

## 1.1 Numerické řešení PDR

Nevýhodou analytického řešení je značná teoretická a technická náročnost a tedy skutečnost, že se nedá vždy použít. Existuje velmi široká třída rovnic, které neumíme analyticky řešit. Tehdy se používají jiné numerické metody řešení.

Cílem je seznámit se základními principy numerických metod řešení parciálních diferenciálních rovnic. Tím je nahrazení analytického řešení této úlohy (funkce více proměnných) přibližnými hodnotami tohoto řešení ve vybraných obvykle předem daných bodech definičního oboru. Tyto body obvykle nazýváme uzly nějaké sítě. Samotné hodnoty v těchto bodech jsou obvykle řešením soustavy jiných rovnic než parciálních diferenciálních. Obvykle se jedná o soustavu lineární (v některých případech i nelineárních algebraických rovnic. Jejich sestavení závisí podstatným způsobem na volbě uzlů sítě. V dalším se omezíme pouze na parciální diferenciální rovnice, kde je řešením funkce dvou proměnných  $(x, y)$ .

Dále se budeme zabývat pouze dvěma možnostmi stanovení sítě a sice na síť obdélníkové, které mají strany rovnoběžné se souřadnými osami a v druhém případě půjde o síť tvořenou trojúhelníky. Další omezení je v řešení některých typů parciálních diferenciálních rovnic druhého řádu. Poznamenejme, že numerické metody je možné použít na nalezení jednoho konkrétního (partikulárního) řešení. Z této skutečnosti vyplývá také požadavek na určení tohoto řešení. To řešení bude určeno okrajovými podmínkami na hranici oblasti, ve které je hledáno. Tyto podmínky mohou být dány předepsáním funkčních hodnot nebo hodnot parciálních derivací tohoto řešení.

V závislosti na volbě sítě budeme hovořit o dvou metodách

- **Metoda konečných diferencí**, kdy jednotlivé parciální derivace nahrazujeme rozdíly funkčních hodnot řešení podobně jako v případě obyčejných diferenciálních rovnic.
- **Metoda konečných prvků**, kdy je zvolená síť tvořena trojúhelníky.

## 1.2 Metoda konečných diferencí pro PDR

Uvažujme modelový případ: lineární parciální diferenciální rovnici 2. řádu eliptického typu:

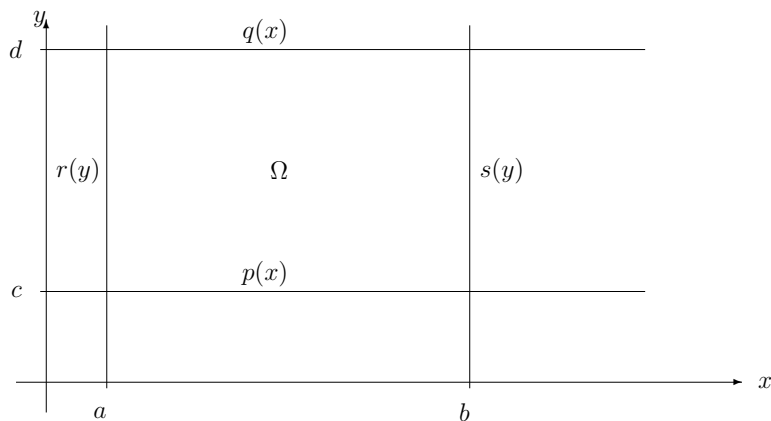
$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} + \sigma(x, y)u = f(x, y), \quad (1.1)$$

kde  $u = u(x, y)$  je definována na oblasti  $\Omega = \{(x, y) : a \leq x \leq b, c \leq y \leq d\}$ , a funkce  $\sigma(x, y) \geq 0 \forall x, y \in \Omega$ ,  $\sigma, f$  jsou spojité na  $\Omega$ .

Nechť je splněna tzv. Dirichletova okrajová podmínka na hranicích oblasti  $\Omega$ :

$$\begin{aligned} u(x, c) &= p(x), & u(x, d) &= q(x), & a \leq x \leq b, \\ u(a, y) &= r(y), & u(b, y) &= s(y), & c \leq y \leq d, \\ p(a) &= r(c), p(b) = s(c), & q(a) &= r(d), q(b) = s(d). \end{aligned}$$

Poslední řádek nám zabezpečuje spojitost okrajových podmínek v „rozích“ oblasti  $\Omega$ . Viz obrázek 1.1.



Obrázek 1.1: Metoda konečných diferencí

Vytvoříme síť na oblasti  $\Omega$  (nejčastěji se používají čtvercové a nebo obdélníkové sítě).

$$x_i = a + ih, \quad i = 0, 1, \dots, n, n+1, \quad h = \frac{b-a}{n+1},$$

$$y_j = c + jk, \quad j = 0, 1, \dots, m, m+1, \quad k = \frac{d-c}{m+1}.$$

Uzly jsou pak body  $(x_i, y_j)$ . Označme  $u_{ij} = u(x_i, y_j)$ . Jestliže předpokládáme spojitost  $u(x, y)$ , pak pro dostatečně malé  $h, k$  můžeme zanedbat chybové funkce. Potom dosazením do (1.1) dostáváme pro  $i = 1, 2, \dots, n, j = 1, 2, \dots, m$ :

$$\frac{2u_{ij} - u_{i+1,j} - u_{i-1,j}}{h^2} + \frac{2u_{ij} - u_{i,j+1} - u_{i,j-1}}{k^2} + \sigma_{ij}u_{ij} = f_{ij}.$$

Pro pravidelné čtvercové sítě, t.j.  $h = k$ , se tato soustava dále zjednoduší na tvar

$$(4 + h^2\sigma_{ij}) u_{ij} - u_{i+1,j} - u_{i-1,j} - u_{i,j+1} - u_{i,j-1} = h^2 f_{ij}. \quad (1.2)$$

Všimněte si, že matice soustavy je v obou případech pro  $\sigma_{ij} \neq 0$  diagonálně dominantní a proto můžeme použít i iterační metody řešení.

Analogický postup můžeme použít pro numerické řešení všech lineárních parciálních diferenciálních rovnic druhého řádu. Postup řešení je vždy stejný: derivace nahradíme diferencemi a hledáme řešení soustavy lineárních algebraických rovnic.

Pro další parciální derivace se používají aproximace:

$$\begin{aligned} \frac{\partial u(x_i, y_j)}{\partial x} &= \frac{u_{i+1,j} - u_{ij}}{h}, & \frac{\partial u(x_i, y_j)}{\partial y} &= \frac{u_{i,j+1} - u_{i,j}}{k}, \\ \frac{\partial u(x_i, y_j)}{\partial x} &= \frac{u_{i,j} - u_{i-1,j}}{h}, & \frac{\partial u(x_i, y_j)}{\partial y} &= \frac{u_{i,j} - u_{i,j-1}}{k}, \end{aligned}$$

$$\frac{\partial u(x_i, y_j)}{\partial x} = \frac{u_{i+1,j} - u_{i-1,j}}{2h}, \quad \frac{\partial u(x_i, y_j)}{\partial y} = \frac{u_{i,j+1} - u_{i,j-1}}{2k}.$$

$$\frac{\partial^2 u(x_i, y_j)}{\partial x \partial y} = \frac{u_{i+1,j+1} - u_{i+1,j-1} - u_{i-1,j+1} + u_{i-1,j-1}}{2h2k}.$$

Problémy při řešení mohou vznikat, pokud oblast  $\Omega$  není obdelníková.

### 1.2.1 Řešený příklad

**Příklad 1.1.** Metodou konečných diferencí řešte okrajovou úlohu

$$u_{xx} + u_{yy} = 8x$$

$$u(x, 0) = x^3, \quad u(0, y) = 0, \quad u(x, y)|_{x^2+y^2=10} = 10x(y+1),$$

kde oblast  $\Omega$  je vnitřní část čtvrtkruhu

$$x \geq 0, \quad y \geq 0, \quad x^2 + y^2 \leq 10.$$

**Řešení.** Rovnici si upravíme na stejný tvar jako (1.1)

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = -8x.$$

Zvolme čtvercovou síť s krokem  $h = 1$ . Potom máme hraniční body

$$\begin{aligned}u(0, 0) = 0, u(1, 0) = 1, u(2, 0) = 8, u(3, 0) = 27, \\u(0, 1) = u(0, 2) = u(0, 3) = 0, \\u(1, 3) = 40, u(3, 1) = 60.\end{aligned}$$

Ještě potřebujeme znát hodnotu v bodě  $P_{2,2}$ . Protože je

$$Q = (2.449; 2), \varphi(Q) = 73.485, \delta = 0.449,$$

lineární interpolací dostaneme

$$u_{2,2} = \frac{1 \cdot 73.485 + 0.449 \cdot u_{1,2}}{1 + 0.449} = 50.697 + 0.310u_{1,2}.$$

Nyní pro 3 vnitřní uzly sestavíme síťové rovnice podle (1.2), přitom hraniční uzly jsou podtrženy.

$$4u_{1,1} - \underline{u_{0,1}} - u_{2,1} - \underline{u_{1,0}} - u_{1,2} = -8,$$

$$4u_{1,2} - u_{1,1} - \underline{u_{1,3}} - \underline{u_{0,2}} - \underline{u_{2,2}} = -8,$$

$$4u_{2,1} - u_{1,1} - \underline{u_{3,1}} - \underline{u_{2,0}} - \underline{u_{2,2}} = -16$$

a pak přidáním odvozeného vztahu pro  $u_{2,2}$  dostaneme soustavu 4 rovnic o čtyřech neznámých. Po úpravě

$$u_{1,1} = \frac{1}{4}(0 + u_{2,1} + 1 + u_{1,2}) - \frac{1}{4}8,$$

$$u_{2,1} = \frac{1}{4}(u_{1,1} + u_{2,2} + 60 + 8) - \frac{1}{4}16,$$

$$u_{1,2} = \frac{1}{4}(0 + 40 + u_{1,1} + u_{2,2}) - \frac{1}{4}8,$$

$$u_{2,2} = 50.697 + 0.310u_{1,2}.$$

Její řešení je pak

$$u_{1,1} = 12.384,$$

$$u_{2,1} = 30.768,$$

$$u_{1,2} = 25.768,$$

$$u_{2,2} = 58.688.$$

Další postup je pak obvyklý, t.j. zmenšíme krok a opakujeme výpočet až se nám odchylky v uzlových bodech ustálí. □



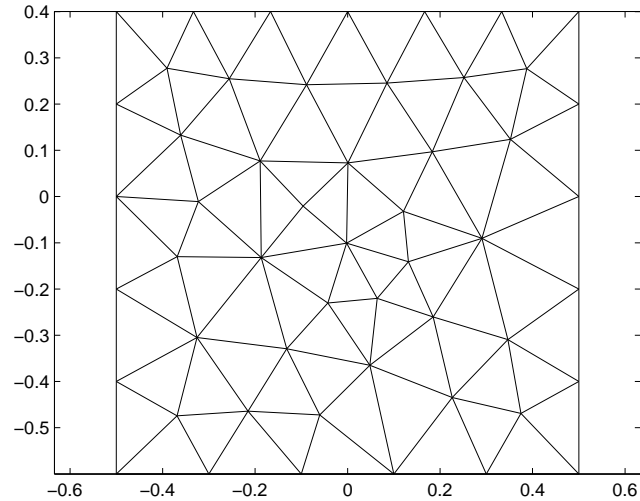
## 1.3 Metoda konečných prvků pro PDR - řešení pomocí Matlabu

Cílem je bez bližšího vysvětlení metody ilustrovat řešení parciální diferenciální rovnice numericky užitím programu Matlab, který má v rámci připravené knihovny rozpracovanou konečných diferencí, ale metodu konečných prvků, která je jednou z nejdůležitějších a nejpoužívanějších numerických metod pro řešení parciálních diferenciálních rovnic.

### 1.3.1 Metoda konečných prvků

Metoda konečných prvků (MKP, anglicky FEM – „finite element method“) má základní přístup podobný jako metoda konečných diferencí (MKD). Opět hledáme řešení parciální diferenciální rovnice na zadané oblasti  $\Omega$ . Matlab umožňuje řešit parciální diferenciální rovnice druhého řádu na rovinných oblastech (tj. hledaná funkce  $u$  závisí na prostorových proměnných  $x$  a  $y$  a případně na čase  $t$ ), ale obecně lze metodou konečných prvků řešit i rovnice vyššího než druhého řádu a ve vyšších dimenzích než v rovině (ovšem už v třírozměrném prostoru se problém technicky velmi zkomplikuje). Všude dál v této kapitole budeme uvažovat pouze rovinné oblasti.

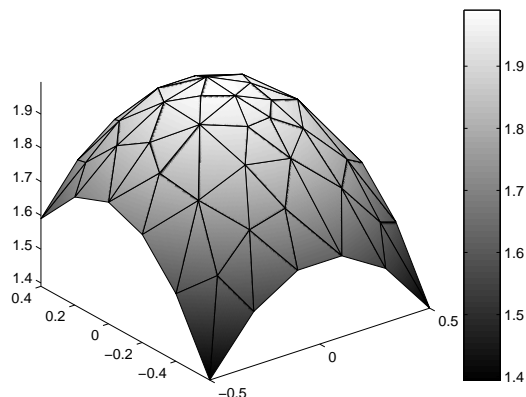
Stejně jako u metody konečných diferencí oblast  $\Omega$  pokryjeme sítí. Tentolrát se však v základní verzi metody konečných prvků používají trojúhelníkové sítě, viz obrázek 1.2. Říkáme, že **oblast  $\Omega$  ztriangulujeme**. Na obrázku vidíme, že síť nemusí být nikterak pravidelná. Je však vhodné, aby pro jednotlivé trojúhelníky nebyly jejich vnitřní úhly nebyly příliš malé.

Obrázek 1.2: Triangulovaná oblast  $\Omega$ 

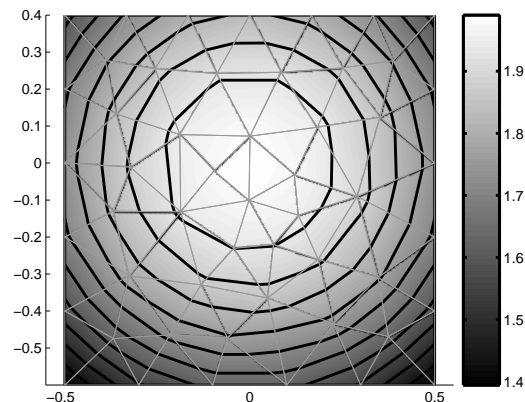
Výhodou trojúhelníkových sítí (oproti obdélníkovým, které se používají u MKD) je to, že mohou dobře vystihnout i velmi složité oblasti. Tím odpadají problémy s realizací okrajových podmínek.

Další analogie s metodou konečných diferencí je v tom, že **původní parciální diferenciální rovnici převedeme na soustavu algebraických rovnic** (lineárních či nelineárních, dle povahy původní úlohy) s neznámými  $u_1, u_2, \dots, u_n$ , kde  $u_i$  je přibližná hodnota řešení v  $i$ -tém uzlu sítě a  $n$  je počet uzlů. Postup, jakým je tato soustava tzv. diskretizačních rovnic získána, je však zcela odlišný a daleko komplikovanější než u MKD a nebude zde popisán. Soustavu diskretizačních rovnic vyřešíme – tím získáme hodnoty řešení v uzlových bodech – a za přibližné řešení rovnice na celé oblasti  $\Omega$  bereme plochu, která uvnitř jednotlivých trojúhelníků nahrazuje řešení kouskem roviny, která prochází funkčními hodnotami ve vrcholech tohoto trojúhelníka. To, že získáme řešení na celé oblasti, a nikoli jen v uzlech sítě, je další výhodou metody konečných prvků oproti MKD.

Pro ilustraci slouží obrázek 1.3, na kterém vidíme přibližné řešení rovnice  $-\Delta u = 4$  na oblasti  $\Omega$  z obrázku 1.2 s okrajovou podmínkou  $u(x, y) = 2 - x^2 - y^2$  na  $\partial\Omega$ . Snadno bychom ověřili, že přesným řešením této rovnice s touto okrajovou podmínkou je funkce  $u(x, y) = 2 - x^2 - y^2$ . Grafem řešení tedy je rotační paraboloid. Vidíme, že přibližné řešení nalezené metodou konečných prvků tvarem paraboloidu vcelku odpovídá, pro přesnější srovnání bychom museli porovnat příslušné numerické hodnoty. Prostorový graf nemusí být ovšem zrovna nejpřehlednější, řešení se proto častěji znázorňuje pomocí vrstevnic, viz obrázek 1.4.



Obrázek 1.3: Graf přibližného řešení



Obrázek 1.4: Přibližné řešení téže rovnice znázorněné pomocí vrstevnic

## 1.4 Příklad řešení pomocí Matlabu

**Poznámka 1.2.** V dalším textu budeme do češtiny míchat různá anglická slova a zacházet s nimi jako s českými. Toto se popisu počítačových programů užívá a je tento přístup vhodnější než důsledný překlad do češtiny.

V Matlabu je pro řešení parciální diferenciální rovnice nainstalovaný tzv. PDE Toolbox (PDE = „partial

differential equations“). Nejpohodlnější je pracovat s grafickým uživatelským rozhraním (GUI). Ukážeme zde řešení jednoho příkladu právě v tomto prostředí.

**Příklad 1.3.** Pomocí Matlabu řešte metodou konečných prvků okrajovou úlohu

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 8x \quad \text{na } \Omega, \quad (1.3)$$

kde oblast  $\Omega$  je čtvrtina kruhu se středem v počátku souřadné soustavy a poloměrem 3, která leží v prvním kvadrantu:

$$\Omega = \{(x, y) : x \geq 0, y \geq 0, x^2 + y^2 \leq 9\},$$

s Dirichletovými okrajovými podmínkami

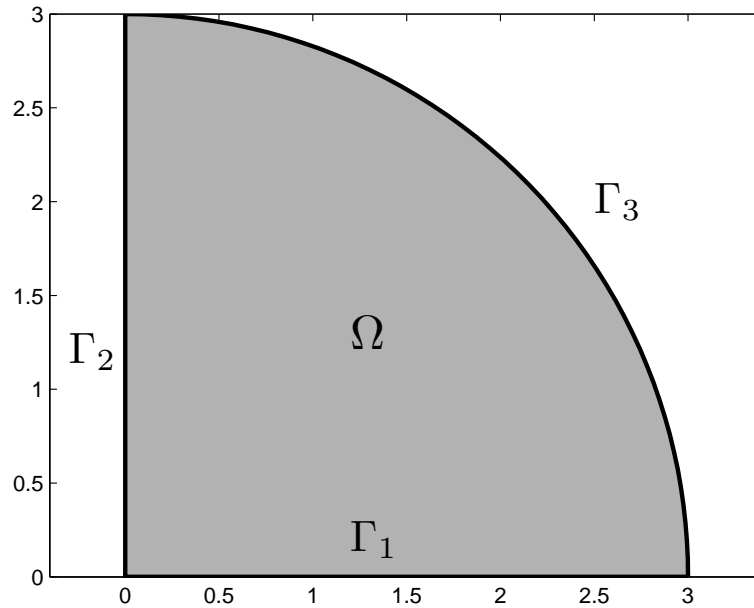
$$\begin{aligned} u(x, y) &= x^3 & \text{na } \Gamma_1 &= \{(x, y) : 0 \leq x \leq 3, y = 0\}, \\ u(x, y) &= 0 & \text{na } \Gamma_2 &= \{(x, y) : x = 0, 0 \leq y \leq 3\}, \\ u(x, y) &= 9x(y + 1) & \text{na } \Gamma_3 &= \{(x, y) : x \geq 0, y \geq 0, x^2 + y^2 = 9\}. \end{aligned} \quad (1.4)$$

Oblast  $\Omega$  s vyznačenými částmi hranice  $\Gamma_1$ ,  $\Gamma_2$  a  $\Gamma_3$  vidíme na obrázku 1.5.

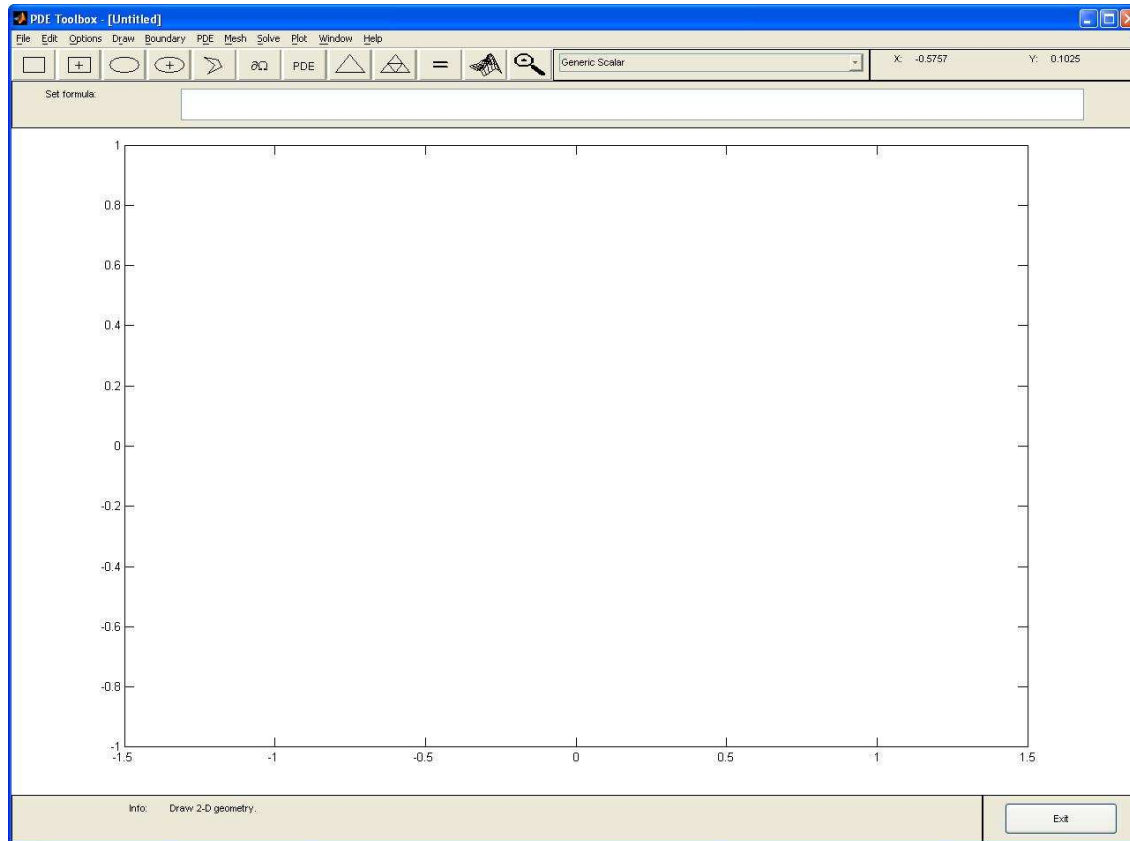
**Řešení.** Spustíme Matlab. Do příkazového okna napíšeme příkaz k otevření prostředí pro řešení parciálních diferenciálních rovnic:

```
>> pdetool
```

Otevře se následující okno



Obrázek 1.5: K příkladu 1.3: Oblast  $\Omega$  a její hranice, rozdělená na tři části



Řešení příkladu v tomto prostředí popíšeme krok za krokem. Každá akce, kterou je třeba provést, bude označena symbolem •.

### Zadání oblasti $\Omega$

Oblast, na které řešíme parciální diferenciální rovnici, můžeme zadat interaktivně. Na bílou plochu uprostřed můžeme umisťovat různé geometrické obrazce (obdélníky, kruhy, elipsy a polygony) a pak z nich pomocí množinových operací (sjednocení, průniku a rozdílu) sestavit požadovanou oblast. Naši oblast (čtvrtkruh) dostaneme jako průnik kruhu se středem v počátku a poloměrem 3 a čtverce, který má levý dolní vrchol v počátku a stranu o délce 3 (případně cokoli většího než 3).

Rozmezí os na ploše neodpovídá našim potřebám, a proto je musíme změnit:

- V menu v položce **Options** vybereme **Axis Limits...** a nastavíme správné rozmezí - pro náš příklad můžeme zadat pro obě osy např. meze -1 až 4.

Dále nastavíme, aby se ukazovala mřížka a aby se body, které budeme za chvíli klikáním zadávat (vrcholy čtverce a pod.) k mřížce přichytávaly.

- V **Options** klikneme na položku **Grid**.
- V **Options** klikneme na položku **Snap** („snap to grid“ = „přilnout k mřížce“).

Nyní zadáme kruh:


- Klikneme na tlačítko se symbolem  $\oplus$ . Tím budeme zadávat elipsu, s tím, že první bod, na který klikneme, je jejím středem.

- Najedeme myší na bod  $(0,0)$ , stiskneme tlačítko myši (díky tomu, že jsme zvolili „snap to grid“, se

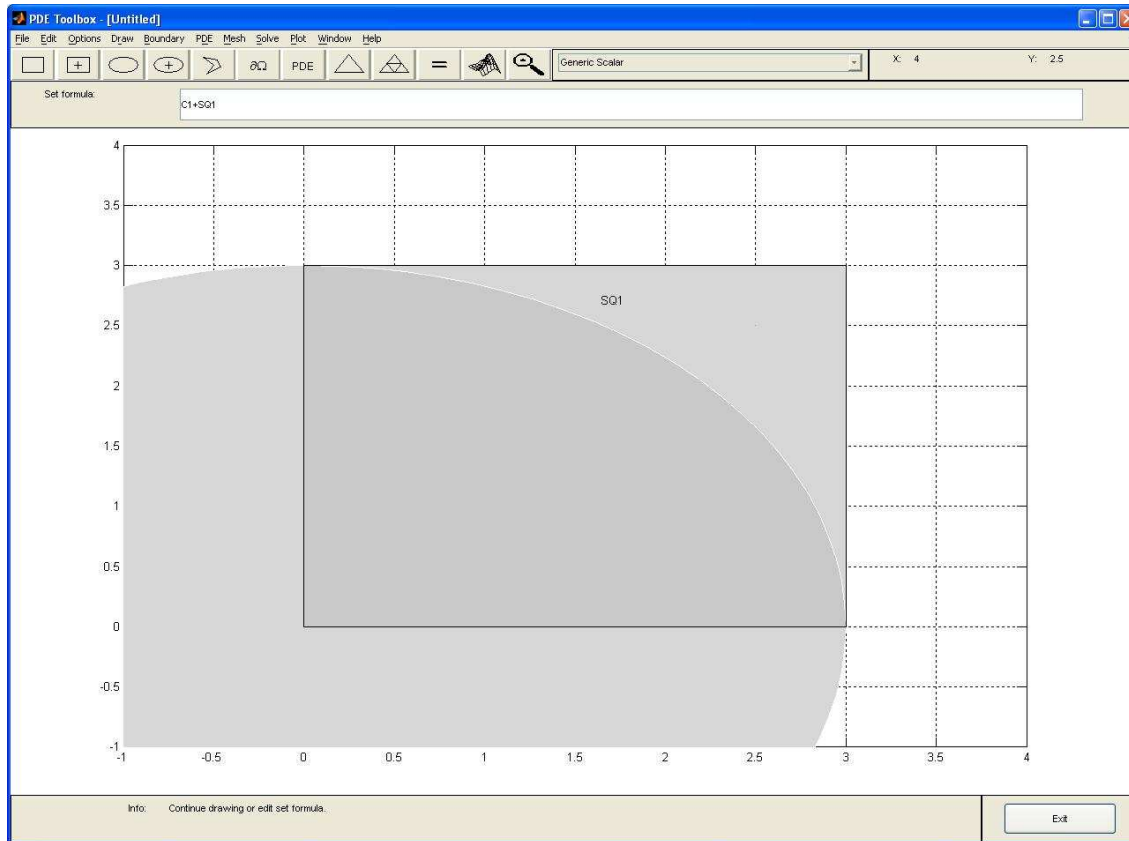


nemusíme trefit úplně přesně) a táhneme - objeví se obrys elipsy. Táhneme, až elipsu (v našem případě vlastně kruh) natáhneme do požadovaných rozměrů.

Podobným způsobem zadáme čtverec:

- Klikneme na tlačítko se symbolem . Tím budeme zadávat obdélník.
- Opět najedeme na bod (0, 0) (levý dolní roh čtverce) a dotáhneme kurzor do bodu (3, 3) (pravý horní roh).

Výsledek by měl vypadat nějak takto:



Kdybychom některý z objektů zadali jinak, než jsme chtěli, můžeme jej vcelku snadno opravit: Nejprve na požadovaný objekt klikneme a tím jej vybereme pro další úpravy (aktuálně vybraný objekt má černě zvýrazněnou hranici). Je-li nevhodně umístěn, ale velikost má přitom správnou, můžeme jej pomocí myši přetáhnout na jiné místo. Chceme-li změnit velikost, stačí, když na vybraný objekt „doubleklikneme“ (Jak je tohle správně česky? Poklepeme? Dvojklikneme?). Otevře se dialog, ve kterém můžeme změnit rozměry, umístění i název. Pokud jsme to zkazili úplně, můžeme vybraný objekt stisknutím klávesy Delete odstranit a začít znovu.

Zatím jsme jen zadali kruh a čtverec, ale nijak jsme počítači nesdělili, že nás zajímá jejich průnik. To provedeme nyní. Podíváme se na políčko nadepsané **Set formula**. Do tohoto políčka zadáváme množinový vzorec („set“ má kromě mnoha jiných i význam „množina“, „formula“ snad překládat netřeba), pomocí něhož je z jednotlivých zadaných oblastí sestavena oblast výsledná. Můžeme používat operátory + (sjednocení), \* (průnik) a – (množinový rozdíl). Při našem zadávání kruhu a čtverce se v poli Set formula automaticky objevil text C1+SQ1. Kdybychom to tak nechali, za oblast  $\Omega$  by se vzalo sjednocení oblastí C1 (C jako „circle“ = kruh) a SQ1 (SQ jako „square“ - čtverec). My ale potřebujeme průnik, a proto

- změníme text v editačním poli **Set formula** na C1\*SQ1. Navenek nepoznáme žádnou změnu, obrázek bude vypadat pořád stejně.

Nyní nastal vhodný okamžik pro uložení výdobytků naší práce.

- Uložíme rozpracovanou úlohu např. jako soubor priklad1.m (klasicky: v menu **File, Save as...**).

Můžeme se do uloženého souboru podívat, např. v editoru Matlabu nebo třeba v Poznámkovém bloku, jak kdo chce. Uvidíme, že Matlab automaticky vytvořil poměrně dlouhý soubor, v němž většině věcí nero-

zumíme, a proto do něj nebudeme vrtat.

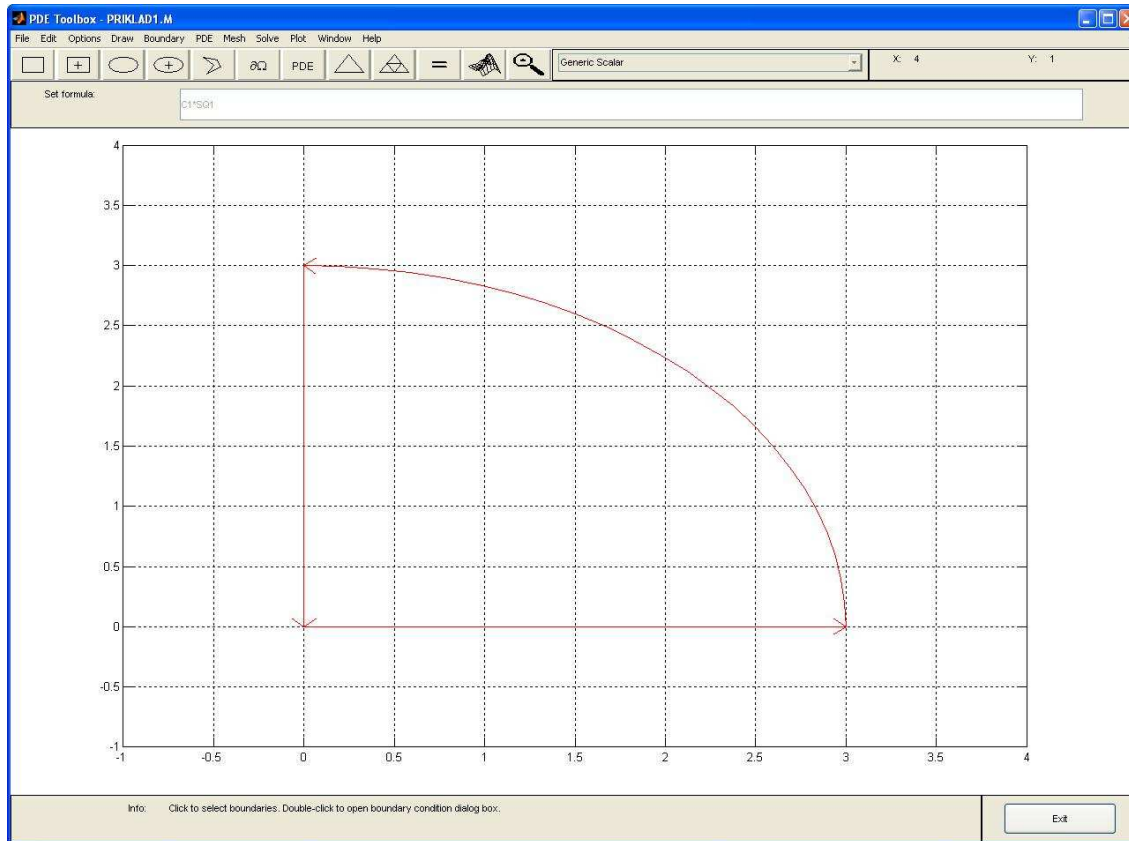
Při další práci je vhodné čas od času úlohu opět uložit, dále již to zdůrazňovat nebudeme.

### Zadání okrajových podmínek

Pokračujeme zadáním okrajových podmínek. Nejprve si necháme zobrazit hranici oblasti.

- Klikneme na tlačítko se symbolem  $\partial\Omega$ . (Jiná možnost: v menu **Boundary**, pak **Boundary Mode**.)

Ukáže se nám hranice oblasti:



Postupně zadáme okrajové podmínky podle předpisů (1.4). Nejprve zadáme podmínku na vodorovné části hranice  $\Gamma_1$  (viz též obrázek 1.5).

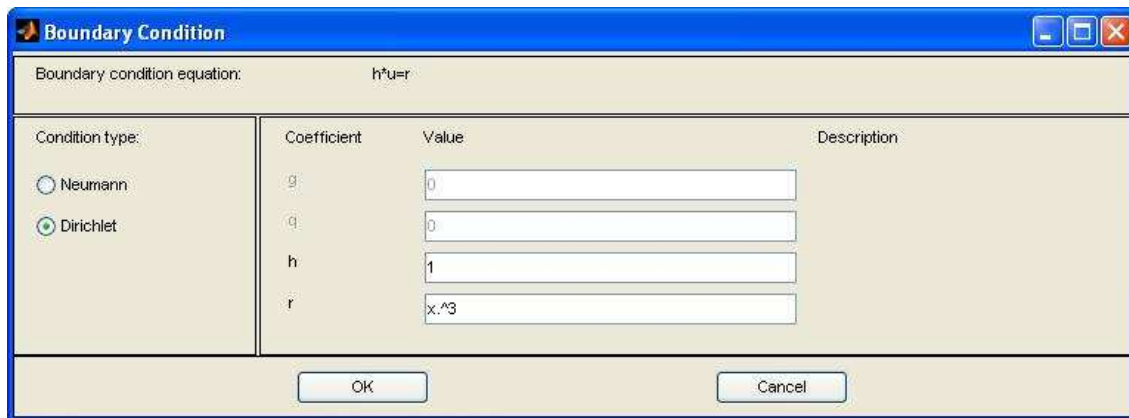
- Doubleklikneme na vodorovnou část hranice. Otevře se dialog pro zadávání okrajové podmínky. Jiná možnost: Na příslušnou část hranice klikneme (jednou). Tím bude tato část hranice vybrána pro další práci. Pak vybereme v menu **Boundary** a **Specify Boundary Conditions...**

V dialogu nyní zadáme okrajovou podmínku  $u(x, y) = x^3$ . V Matlabu lze zadávat dva základní druhy okrajových podmínek, Dirichletovy (je přímo zadáno, čemu se má řešení na hranici rovnat) a Neumannovy (které obsahují též derivaci řešení ve směru normály k hranici). Naše okrajové podmínky jsou Dirichletova typu.

- V dialogu proto vybereme (či spíš necháme nastaveno) **Condition type** na **Dirichlet**.

Matlab očekává nyní podmínku ve tvaru (viz horní část dialogu)  $h \cdot u = r$ , kde  $h$  a  $r$  jsou zadané funkce, případně konstanty. Pro naši podmínku  $u(x, y) = x^3$  je  $h = 1$  a  $r(x, y) = x^3$ .

- V dialogu vyplníme kolonku pro funkci  $r$  výrazem  $x.^3$ . (Funkce  $h$  je na jedničku nastavená automaticky.) Vyplněný dialog by měl vypadat takto:



Podobným způsobem zadáme okrajové podmínky na ostatních částech hranice:

- Doubleklikneme na svislou část hranice. V dialogu zkontrolujeme, zda je zatržen Dirichletův typ okrajových podmínek a vyplníme kolonku pro funkci  $r$  výrazem 0.
- Totéž provedeme pro obloukovou část hranice, tentokrát zadáme  $9 \cdot x \cdot (y+1)$ .

### Zadání rovnice

Nyní zadáme samotnou parciální diferenciální rovnici, kterou chceme vyřešit.

- Klikneme na tlačítko **PDE** nebo v menu vybereme **PDE** a pak **PDE Specification...**

Otevře se dialog pro zadání rovnice. V levé části dialogu vybíráme typ rovnice - na výběr máme rovnici eliptickou, parabolickou, hyperbolickou a problém vlastních čísel. Naše rovnice (1.3) je typu eliptického, proto

- vybereme (případně jen zkontrolujeme, zda je vybrán) z možností pro **Type of PDE** typ **Elliptic**. Rovnice je nyní očekávána (viz horní část dialogu) ve tvaru

$$-\operatorname{div}(c \cdot \operatorname{grad} u) + au = f \quad (1.5)$$

a po nás se chce, abychom doplnili funkce  $c$ ,  $a$  a  $f$ . Doufáme, že tvarem (1.5) nejste příliš zaskočení, pro jistotu však připomeňme, že je-li  $f$  funkce proměnných  $x_1, x_2, \dots, x_n$ , pak gradient funkce  $f$  je

$$\operatorname{grad} f = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right),$$

a jsou-li  $g_1, g_2, \dots, g_n$  funkce proměnných  $x_1, x_2, \dots, x_n$ , pak divergence zobrazení  $g = (g_1, \dots, g_n)$  je

$$\operatorname{div} g = \frac{\partial g_1}{\partial x_1} + \frac{\partial g_2}{\partial x_2} + \dots + \frac{\partial g_n}{\partial x_n}.$$

Těž si snad pamatujete, že

$$\operatorname{div}(\operatorname{grad} f) = \frac{\partial}{\partial x_1} \left( \frac{\partial f}{\partial x_1} \right) + \dots + \frac{\partial}{\partial x_n} \left( \frac{\partial f}{\partial x_n} \right) = \frac{\partial^2 f}{\partial x_1^2} + \dots + \frac{\partial^2 f}{\partial x_n^2} = \Delta f.$$



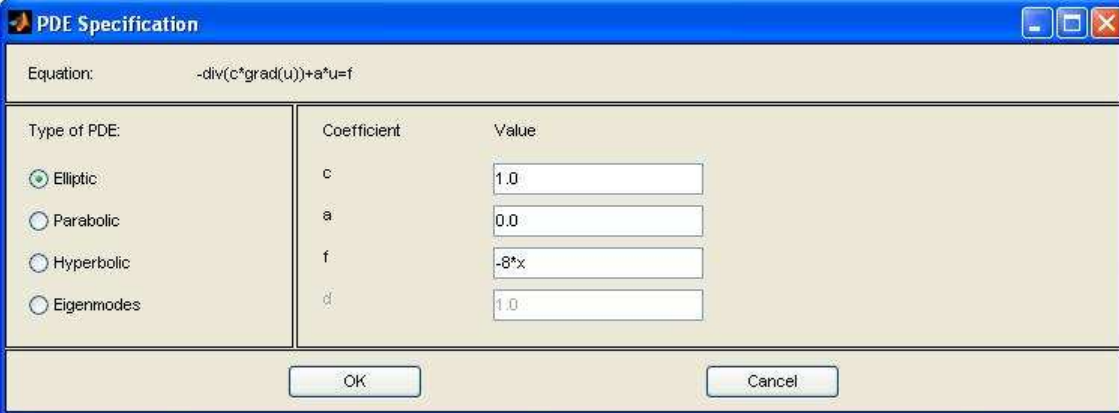
Řešená rovnice (1.3), tj.  $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 8x$ , po snadné úpravě  $-\Delta u = -8x$ , se tedy do tvaru (1.5) přepíše jako

$$-\operatorname{div}(1 \cdot \operatorname{grad} u) + 0 \cdot u = -8x.$$

Proto dialog pro zadání rovnice doplníme takto (některé z uvedených hodnot už tam možná jsou „samy od sebe“, pak je pochopitelně necháme být):

- Do kolonky pro funkci  $c$  doplníme konstantu 1, do kolonky pro  $a$  napíšeme nulu a do kolonky pro  $f$  napíšeme  $-8*x$ .

Celá věc by pak měla vypadat následovně:



The screenshot shows a dialog box titled "PDE Specification". At the top, the equation is given as  $-\operatorname{div}(c \cdot \operatorname{grad}(u)) + a \cdot u = f$ . Below this, there are four radio buttons for the "Type of PDE": Elliptic (selected), Parabolic, Hyperbolic, and Eigenmodes. To the right, there are four input fields for coefficients:  $c$  (1.0),  $a$  (0.0),  $f$  (-8\*x), and  $d$  (1.0). At the bottom, there are "OK" and "Cancel" buttons.

## Triangulace oblasti

Oblast  $\Omega$  ztriangulujeme:

- Klikněte na tlačítko s trojúhelníkem nebo v menu vyberte **Mesh** a pak **Initialize Mesh**

Chceme-li mít síť jemnější, můžeme

- kliknout na tlačítko s trojúhelníkem rozděleným na čtyři menší trojúhelníky nebo v menu vybrat **Mesh** a pak **Refine Mesh**.

Zjemněnou síť pak můžeme ještě poněkud vylepšit (zpravidelnit, odstranit z ní některé úzké trojúhelníky) pomocí funkce **Jiggle**. (Původní význam slova „jiggle“ je „pohupovat“ či „trhaně se pohybovat“, v souvislosti se sítí se tím myslí zhruba to, že uzly sítě se trochu popřemísťují, každý uzel se přesune do průměru svých sousedů.)

- Můžeme v menu vybrat **Mesh** a pak **Jiggle Mesh** - tuto operaci můžeme provádět i opakovaně.

Pokud se nám to, co se se sítí děje, nelíbí, můžeme úpravy brát zpět pomocí menu: **Mesh**, pak **Undo Mesh Change**. První návrh sítě můžeme ovlivnit pomocí nastavení parametrů (v menu **Mesh**, **Parameters...**). Zde můžeme např. nastavit maximální povolenou délku hrany (**maximum edge size**).

Jestliže jsme zjemnění a „jigglování“ provedli právě jednou, měli bychom mít na obrazovce toto:



Síť můžeme exportovat pro její případné další použití: v menu Mesh, pak Export Mesh... Síť bude uložena pomocí tří matic, jejichž jména si můžeme vybrat. Matlab nám nabízí jména **p**, **e** a **t**. První matice bude obsahovat informace o uzlech sítě (points), druhá o hranách (edges) a třetí o trojúhelnících (triangles). S těmito maticemi pak v Matlabu můžeme dále pracovat dle potřeby, např. si jejich prvky můžeme nechat vypsát do souboru, který pak můžeme přenést a používat v jiném programu.

### Řešení rovnice a jeho grafické znázornění

Teď když máme všechno připravené, můžeme konečně najít přibližné řešení rovnice.

- Klikneme na tlačítko s rovnítkem nebo v menu vybereme **Solve**, pak **Solve PDE**.

Řešení bude asi chvilku trvat (je nutno ne zcela jednoduchým způsobem sestavit a pak vyřešit systém zhruba 500 rovnic o 500 neznámých – pokud pracujeme se sítí, která je na předchozím obrázku). Až je počítač hotov, řešení se zobrazí pomocí dvourozměrného obrázku – hodnoty řešení na oblasti  $\Omega$  jsou rozlišeny pomocí barev, vpravo máme zobrazenou stupnici (angl. colorbar), pomocí níž poznáme, jaká barva odpovídá jaké hodnotě řešení.

Chceme-li řešení exportovat pro další použití, uděláme to přes menu: Solve, pak Export Solution... Řešení je uloženo ve formě vektoru (jméno si můžeme vybrat, automaticky se nabízí **u**), jehož složky jsou hodnoty řešení v jednotlivých uzlech sítě. Chceme-li s tímto řešením dále pracovat, případně je (přes soubor) přenést do nějakého jiného programu, musíme k němu samozřejmě mít i příslušnou síť, hlavně její uzly, jinak je zcela bezcenné.

Parametry zobrazení řešení můžeme měnit. Formulář pro nastavení parametrů se nám zobrazí po stisk-

nutí tlačítka s 3-D grafem, případně se k němu dostaneme z menu: Plot, Parameters... Zde již necháme každému čtenáři prostor pro experimentování.  $\square$

## Cvičení

1. Pomocí Matlabu najděte přibližné řešení úlohy

$$-\Delta u = 5 \sin\left(10 \arctg \frac{y}{x}\right) \quad \text{na } \Omega,$$

kde  $\Omega$  je část mezikruží o poloměrech  $r = 1$ ,  $R = 3$ , která leží v prvním kvadrantu,

$$\Omega = \{(x, y) : x \geq 0, y \geq 0, 1 \leq x^2 + y^2 \leq 9\},$$

s okrajovými podmínkami

$$\begin{aligned} u(x, y) &= 0 & \text{na } \Gamma_1 &= \{(x, y) : x \geq 0, y \geq 0, x^2 + y^2 = 1\}, \\ u(x, y) &= 5 & \text{na } \Gamma_2 &= \{(x, y) : x \geq 0, y \geq 0, x^2 + y^2 = 9\}, \\ \vec{n} \cdot \text{grad } u &= 0 & \text{na } \Gamma_3 &= \{(x, y) : 1 \leq x \leq 3, y = 0\} \text{ a } \Gamma_4 = \{(x, y) : x = 0, 1 \leq y \leq 3\}. \end{aligned}$$

## Výsledky

### 1. Spíše několik poznámek a tipů:

Pozor na okrajovou podmínku zadanou na  $\Gamma_3$  a  $\Gamma_4$ . Symbolem  $\cdot$  se zde nemyslí obyčejné násobení, ale skalární součin. Podmínka mohla být též zapsána jako  $\partial u / \partial \vec{n} = 0$ . Jedná se o homogenní Neumannovu okrajovou podmínku, které se též říká podmínka kolmosti. Až budete mít řešení, nechte si zobrazit vrstevnice (contours) a pokuste se odhadnout, proč se v souvislosti s touto podmínkou zmiňuje zrovna kolmost.

Při zadávání této podmínky si můžete všimnout, že v Matlabu bude očekáván tvar  $\mathbf{n} * \mathbf{c} * \text{grad}(u) + \mathbf{q}u = \mathbf{g}$ . Písmem  $\mathbf{n}$  se myslí normála  $\vec{n}$ , takže nezadáváme nic. Funkce  $c$  je tatáž jako v rovnici (viz (1.5)) a zadáme ji (nebo spíš necháme nastavenou na 1) až při zadávání rovnice. Jediné, co musíme zadat přímo zde, jsou funkce  $q$  a  $g$ . Pravá strana rovnice je poněkud komplikovaná, ale není to ze zlomyslnosti, spíš k tomu vedla snaha o to, aby výsledný obrázek byl zajímavější než u řešeného příkladu. Pozor na správný zápis operátorů - nezapomeňte na patřičné místo napsat tečku. Funkce arctg se v Matlabu zadává jako `atan`, ne třeba `arctan`! Z toho, že zlomek  $y/x$  na části hranice oblasti  $\Omega$  není definován, si nemusíte dělat hlavu - Matlab si s tím poradí, zvláště, když se tento zlomek dále dosazuje do funkce arctg.