

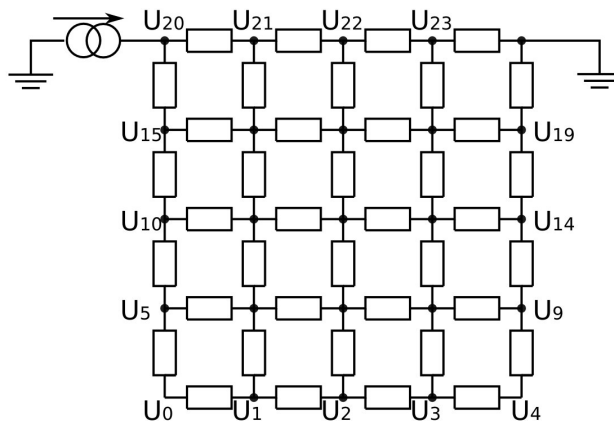
Zadání příkladů pro cvičení z předmětu Programování pro fyziky

Úloha č. 8 — 3. prosince 2013

Nejprve pro nedočkavé: pro úspěšné vyřešení úlohy nemusíte číst motivační část, lze začít rovnicí (1). Komu přijdou obvody nudné, může zkusit <http://utf.mff.cuni.cz/~ledvinka/CvProgFyz/2019-4.pdf>, kde se počítají rezonanční obrazce.

Uvažujte čtvercovou síť $N \times N$ z jednotkových odporů podle Obrázku 1. Napište program, který řešením soustavy lineárních rovnic nalezne hodnotu elektrického odporu jakou naměříte mezi sousedními vrcholy čtverce.

Při měření odporu mezi dvěma uzly jeden uzemníme a druhý připojíme ke zdroji jednotkového proudu. Napětí v tomto uzlu je pak rovno měřenému odporu. Bohužel toto napětí nelze snadno spočítat bez znalosti napětí v ostatních uzlech obvodu. Proto napětí U_0 až U_{N^2-2} tvoří neznámé veličiny, pro které je třeba sestavit soustavu rovnic. Ta je dána Kirchhoffovým zákonem, který říká, že součet proudů vtékajících do každého uzlu je nula. Povšimněme si tvaru těchto rovnic v některých typických bodech (pro jednoduchost užíváme $N = 5$). Nejprve vnitřní body sítě



Obrázek 1. Síť odporů pro $N = 5$.

$$\frac{U_6 - U_5}{R} + \frac{U_6 - U_7}{R} + \frac{U_6 - U_1}{R} + \frac{U_6 - U_{11}}{R} = 0.$$

Na stranách čtverce a v jeho rozích platí

$$\frac{U_2 - U_1}{R} + \frac{U_2 - U_3}{R} + \frac{U_2 - U_7}{R} = 0, \quad \frac{U_4 - U_3}{R} + \frac{U_4 - U_9}{R} = 0.$$

Výjimku tvoří body blízko přípojek pro měření odporu

$$\frac{U_{20} - U_{15}}{R} + \frac{U_{20} - U_{21}}{R} = I, \quad \frac{U_{19} - U_{18}}{R} + \frac{U_{19} - U_{14}}{R} + \frac{U_{19}}{R} = 0, \quad \frac{U_{23} - U_{22}}{R} + \frac{U_{23} - U_{18}}{R} + \frac{U_{23}}{R} = 0.$$

Všimněte si, že to, jestli leží uzel k na straně čtverce, jeho rohu, nebo uvnitř lze zjistit z hodnoty výrazů $i = k \bmod N$ a $j = k \div N$

1. Vyjděte z příkladu na webu cvičení, který řeší soustavu lineárních rovnic

$$Ax = b$$

Gauss-Jordanovou eliminací <http://utf.mff.cuni.cz/~ledvinka/?229760>. Můžete samozřejmě použít i jiný kód. **Pozor:** Kromě kódu ze cvičení naleznete i doplněnou verzi, kde je matice i pravá strana obsazována funkcemi `maticeSoustavy` a `pravaStrana`. Kód naleznete pod názvem `cv10_eliminate2.pas` resp. `cv10_eliminate2.cpp`.

2. Napište proceduru, která obsadí matici A_{ij} soustavy

$$\sum_j A_{ij} U_j = I_i \quad (1)$$

a její pravou stranu I_i (u níž je nenulový jen jeden prvek), v souladu s obvodem na Obr. 1 a Kirchovovými zákony. Indexy $i, j = 0..N^2 - 2$ a pro hodnotu všech odporů $R = 1$ je

$$A_{ij} = \begin{cases} 4 & i = j \text{ a bod } \{i\} \text{ leží uvnitř} \\ 3 & i = j \text{ a bod } \{i\} \text{ leží na straně} \\ 2 & i = j \text{ a bod } \{i\} \text{ leží na rohu} \\ -1 & \text{z bodu } \{i\} \text{ do } \{j\} \text{ vede odpor} \\ 0 & \text{jinak} \end{cases} \quad (2)$$

Pozn. napětí v uzemněném uzlu není neznámé a jeho uzemněnost rovnice (2) uvažuje.

3. Zaříd'te, aby výstup programu měl pro $N = 3$ tuto podobu

$$\begin{array}{cccccccc|c} 2 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 3 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 2 & -1 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 3 & 0 \end{array}$$

Odpor = 1.2500

a aby vypadal analogicky i pro $3 < N \leq 22$.

4. Spočt'ete odpor pro $N = 22$.

Kód Vašeho programu odevzdejte na obvyklé webové stránce ve stavu, kdy počítá řešení pro $N = 22$. Hodnotu N v programu deklarujte jako konstantu, kterou lze změnit, aniž by program přestal fungovat (sami vyzkoušejte, že program běží i pro hodnoty $N = 3$ a $N = 4$.) **Narazíte-li při řešení na problémy**, nejprve znovu z webu stáhněte aktuální verzi zadání, je možné, že v něm bude již zanesena odpověď na vaši otázku.

Dodatek: Při testování počtu rezistorů, které vycházejí z nějakého uzlu, lze v rovnici (2) použít explicitní (případně implicitní v C++) konverzi logické hodnoty na celé číslo

```
A[i,i] = 2 + ord((radek>0) and (radek<N-1)) + ord((sloupec>0) and (sloupec<N-1));
```

resp.

```
A[i][i] = 2 + (radek>0 && radek<N-1) + (sloupec>0 && sloupec<N-1);
```