Table of Contents

(DÚ4) Optimální proložení bodů kružnicí

Mějme m bodů v rovině, $\mathbf{a}_1, \ldots, \mathbf{a}_m \in \mathbb{R}^2$. Chceme najít kružnici se středem $\mathbf{c} \in \mathbb{R}^2$ a poloměrem $r \geq 0$ takovou, že součet čtverců vzdáleností bodů od kružnice je nejmenší.

Označme jako $\mathbf{x} = (\mathbf{c}, r) \in \mathbb{R}^3$ vektor parametrů kružnice. Nechť $\operatorname{dist}(\mathbf{x}, \mathbf{a})$ je orientovaná vzdálenost bodu $\mathbf{a}$ od kružnice s parametry $\mathbf{x}$. Tedy $|\operatorname{dist}(\mathbf{x}, \mathbf{a})|$ je Eukleidovská vzdálenost bodu $\mathbf{a}$ od kružnice, přičemž pro $\mathbf{a}$ vně kružnice je $\operatorname{dist}(\mathbf{x}, \mathbf{a}) > 0$ a pro $\mathbf{a}$ uvnitř kružnice je $\operatorname{dist}(\mathbf{x}, \mathbf{a}) < 0$. Chceme minimalizovat funkci

$$f(\mathbf{x}) = \sum_{i=1}^{m} \operatorname{dist}(\mathbf{x}, \mathbf{a}_i)^2$$

Implementační úkoly

1. Implementujte funkci `d = dist(x, A)`, kde `x` je vektor 3×1, `A` je matice $2 \times m$ obsahující body $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m$, `d` je vektor $N \times 1$.
2. Implementujte funkci `[x_new] = make_GN_iter(x, A)`, která provede jednu iteraci čisté (tedy s jednotkovou délkou kroku) Gauss-Newtonovy metody.
3. Implementujte funkci `[x_new, success] = make_LM_iter(x, A, mu)`, která provede jednu iteraci Levenberg-Marquardtovy metody (bližší popis funkcí je v jejich šablonách.)

Poznámky:

- Pro práci na úloze máte k dispozici skript `main.m` a funkci `fit_circle.m`.
- Udělejte si v matlabu jednoduchý skript, který vám dovolí naklikat body $\mathbf{a}_i$ a uložit si je do MAT souboru (pro naklikání se bude hodit funkce `ginput`.) Takto si připravte několik zajímavých konfigurací bodů, mezi kterými pak můžete snadno přepínat a zkoušet na nich svoje implementace algoritmů.

Teoretické úkoly (do zprávy)

1. Mějme několik bodů $\mathbf{a}_1, \ldots, \mathbf{a}_m$ v obecné konfiguraci. Je funkce f všude diferencovatelná? Má jedno nebo více lokálních minim? Odpovědi zdůvodněte. Navrhujeme (ale nemusíte to dělat), abyste při vyšetřování funkce $f(\mathbf{x}) = f(\mathbf{c}, r)$ uvažovali funkci $h(\mathbf{c}) = f(\mathbf{c}, r)$ pro nějaká konstantní r, tedy řez funkce f podle r. Grafy takových řezů lze snadno vizualizovat.
2. Diskutujte, jaký algoritmus je vhodný na minimalizaci funkce $f(\mathbf{x})$ a proč. Čím více myšlenek a argumentů uvedete, tím lépe. Je možné, aby Gaussův-Newtonův algoritmus na naší úloze divergoval?
3. Může se zdát, že algoritmy na nelineární nejmenší čtverce bez omezení nejde použít, protože máme omezení $r \geq 0$. Vadí to? Co se stane, budeme-li toto omezení ignorovat? Můžou algoritmy konvergovat k řešení se záporným r? Své odpovědi odůvodněte.
4. Najděte nějakou množinu $m \geq 3$ bodů $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m\}$ a takovou dvojici počátečních parametrů kružnice $\mathbf{x}_0^{(1)}$ a $\mathbf{x}_0^{(2)}$, aby algoritmus inicializovaný těmito parametry skončil v různých lokálních minimech. Do zprávy udělejte následující tabulku s obrázky (všechny elementy tabulky jsou obrázky exportované z matlabu, např. pomocí funkce `print`):

body a kružnice $\mathbf{x}_0^{(1)}$	body a stav dokonvergovaný z $\mathbf{x}_0^{(1)}$	graf `f_history` (kritérium v závislosti na indexu iterace) pro $\mathbf{x}_0^{(1)}$
body a kružnice $\mathbf{x}_0^{(2)}$	body a stav dokonvergovaný z $\mathbf{x}_0^{(2)}$	graf `f_history` (kritérium v závislosti na indexu iterace) pro $\mathbf{x}_0^{(2)}$

Co potřebujete udělat:

- Z gitlabu si stáhněte šablony pro funkce k implementaci a pomocné funkce a skripty.
- Implementujte funkce popsané výše.
- Napište PDF zprávu a pojmenujte ji `report.pdf`
- Zabalte všechny implementované funkce (plus jakékoli vaše pomocné funkce, které výše čtyři zmíněné používají) a PDF zprávu do ZIP souboru a nahrajte je do upload systému. Udělejte ZIP soubor tak, aby se vaše soubory rozbalily rovnou do aktuálního adresáře, ne do nějakého podadresáře (jinak to nebude fungovat.)