Niclas råd vid lösning av verkligt minkva-problem

1. Härled uttryck för modellen $g(x)$, t.ex.

$$g(x) = \begin{bmatrix} c_x \\ c_y \end{bmatrix} + \begin{bmatrix} \cos\delta & -\sin\delta \\ \sin\delta & \cos\delta \end{bmatrix} \begin{bmatrix} a\cos\theta \\ b\sin\theta \end{bmatrix},$$

$$x = [c_x, c_y, a, b, \delta, \theta] .$$

2. Implementera $g(x)$, i vektorform, d.v.s. ett anrop genererar flera mätpunkter, t.ex.

$$G(x) = \begin{bmatrix} g(x_1) & g(x_2) & \dots & g(x_m) \end{bmatrix}$$

eller

$$G(x) = \begin{bmatrix} g(x_1) \\ g(x_2) \\ \vdots \\ g(x_m) \end{bmatrix},$$

$$x_i = [c_x, c_y, a, b, \delta, \theta_i] .$$

   (a) Verifiera $g(x)$ m.h.a realistiska parametervärden.

3. Konstruera och implementera residualen $F(x)$ som “modell minus data”

$$F(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_m(x) \end{bmatrix},$$

$$f_i(x) = g(x) - p_i,$$

där p_i är mätdata.

   (a) Verifiera att $F(x_*) = 0$, dvs generera p_i m.h.a. $G(x)$.

4. Härled uttryck för jacobianen $J(x)$. Använd gärna verktyg för symboliskt derivering, t.ex. Maple. Implementera jacobianen.

   (a) Verifiera den symboliska jacobianen med en numerisk approximation:

$$J(x) \approx \begin{bmatrix} \frac{F(x+\epsilon_1)-F(x)}{\epsilon} & \dots & \frac{F(x+\epsilon_n)-F(x)}{\epsilon} \end{bmatrix},$$

$$\epsilon_1 = [\epsilon, 0, \dots, 0]^T, \ \epsilon_n = [0, \dots, 0, \epsilon]^T.$$

5. Generera mätdata utan fel, dvs $p_i = g(x_*)$.

   (a) Kontrollera att $F(x_*) = 0$ och att $J(x_*)$ har full rang.

   (b) Kontrollera att er algoritm ger x_* som lösning efter max 1 iteration då $x_0 = x_*$.

   (c) Stör startapproximationen x_0 och kontrollera att ni får konvergens mot x_* från rimligt stort område.

6. Generera mätdata med liten störning ortogonal mot jacobianen i lösningen, $p_i = g(x_*) + \epsilon$, $J^T\epsilon = 0$.

   (a) Kontrollera att $F(x_*) \neq 0$ och att $J(x_*)^T F(x_*) = 0$.

   (b) Kontrollera att er algoritm ger x_* som lösning efter max 1 iteration då $x_0 = x_*$.

   (c) Stör startapproximationen x_0 och kontrollera att ni får konvergens mot x_* från rimligt stort område.

   (d) Ev. upprepa 6c med större fel.

7. Kontrollera störningskänsligheten hos lösningen genom att upprepa följande

   (a) Generera mätdata med liten störning, $p_i = g(x_*) + \epsilon$.

   (b) Lös optimeringsproblemet med $x_0 = x_*$. Kalla lösningen på det störda problemet $\tilde{x}_*$.

   (c) Jämför lösningen på det störda problemet $\tilde{x}_*$ med lösningen på det ostörda x_*.

8. Upprepa ev. ovan med störd x_0.

9. Ta fram funktion för startapproximation som endast baseras på mätdata.

10. Kontrollera kvalitén på startapproximationen genom att

    (a) Generera mätdata utan störning.

    (b) Bestäm x_0 och lös optimeringsproblemet med $x_0 = x_*$. Kalla lösningen på det störda problemet $\tilde{x}_*$.

    (c) Jämför lösningen på det störda problemet $\tilde{x}_*$ med lösningen på det ostörda x_*.

11. Upprepa 10a–10c med liten resp. realistisk störning.

12. Upprepa 10a–10c på verkliga data.