# Laboration 1: Icke-linjär parameterestimering

Niclas Börlin

11 november 1999

## 1 Inledning

I denna laboration ska ni implementera Gauss-Newtons och Levenberg-Marquardts metod och applicera dessa på ett verkligt problem.

## 2 Implementation

### 2.1 Linjesökning

Implementera en funktion som utför linjesökning från en punkt $x_k$ längs en sökriktning $p_k$ och använder Armijos villkor och backtracking med $\alpha = 1, \frac{1}{2}, \frac{1}{4}, \ldots$.

Förslag på lämpliga inparametrar:

- Namnet på residualfunktionen $F$.
- Nuvarande punkt $x_k$.
- Sökriktningen $p_k$.
- Nuvarande residual $F(x_k)^*$.
- Kortaste accepterade steglängd $\alpha_{\text{min}}$. Nödvändig för att garantera terminering av steglängdsalgoritmen.
- Parameter(-rar) för att kunna kontrollera Armijos villkor.

Förslag på lämpliga returparametrar:

- Längsta accepterade steglängd $\alpha_k$. Om ingen sådan finns $\geq \alpha_{\text{min}}$ skall $\alpha_k = 0$ returneras.
- Nästa punkt $x_{k+1}{}^*$.
- Funktionsvärdet i nästa punkt $f(x_{k+1})^*$.

$^*$-märkta parametrar finns med för att undvika att upprepa beräkningar i onödan och kan uteslutas för att få en ”renare” implementation.

### 2.2 Gauss-Newtons metod

Implementera Gauss-Newtons metod för lösning av icke-linjär minsta-kvadratproblem utan bivillkor. Använd termineringsvillkoret $\|Jp\| \leq \epsilon(1 + \|F\|)$.

Förslag till inparametrar:

- Namnet på residualfunktionen $F$ och Jacobianen $J$. Tips! Om funktionerna ges systematiska namn, t ex `p1_f` (residualen), `p1_j` (jacobianen) behöver bara inledningen `'p1'` skickas.
- En startpunkt $x_0$.
- Maximalt antal tillåtna iterationer.
- Konvergenstoleransen $\epsilon$.
- Parametrarna $\alpha_{min}$ och $\mu$ till steglängdsalgoritmen.

Förslag till utparametrar:

- Förslag till lösning $x_*$.
- Antal iterationer som krävdes.
- En kod som anger om metoden konvergerat, avbrutits ($\alpha \geq \alpha_{min}$ ej funnits) eller divergerat (maximalt antal iterationer förbrukats).
- En matris med samtliga $x_i$ lagrade kolumnvis. Kan vara bra för att återskapa iterationssekvensen för analys.
- En vektor med steglängder. Också bra för att kunna analysera metoden.

### 2.3 Levenberg-Marquardts metod

Implementera Levenberg-Marquardts metod. Använd utdelad funktion `dogleglsq` för att lösa delproblemet

$$
\begin{aligned}
\min_p \psi_k(p) &= f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T \nabla^2 f(x_k) p \\
&\text{s.t.} \\
\|p\| &\leq \Delta_k.
\end{aligned}
$$

Använd $\|Jp_{GN}\| \leq \epsilon(1 + \|F\|)$ som termineringsvillkor.

Förslag till inparametrar:

- Namnet på residualfunktionen $F$ och jacobianen $J$.
- En startpunkt $x_0$.
- Maximalt antal tillåtna iterationer.
- Konvergenstoleransen $\epsilon$.
- Inital trust-region-storlek $\Delta_0$.
- Trust-regions-uppdateringsgränserna $\mu$ (my) och $\eta$ (eta).

Förslag till utparametrar:

- Förslag till lösning $x_*$.
- Antal iterationer som krävdes.
- En kod som anger om metoden konvergerat eller divergerat (maximalt antal iterationer förbrukats).
- En matris med samtliga $x_i$ lagrade kolumnvis. Kan vara bra för att återskapa iterationssekvensen för analys.
- En vektor med steglängder ($\|p_k\|$), konvergensvärden ($\rho_k$) och/eller trust-region-storlekar $\Delta_k$. Också bra för att kunna analysera metoden.

## 2.4 Problemfunktioner

Implementera följande funktioner för ellipsproblemet:

- En funktion som genererar mätdata med simulerade normalfördelade fel givet en parametervektor $x = [c_x, c_y, a, b, \delta, \theta_i]$ och en standardavvikelse $\sigma$.
- En funktion som beräknar residualvektorn $F(x)$.
- En funktion som beräknar jacobianen $J(x)$.

Ni får använda globala (eller statiska) variabler för att lagra "mätpunkterna" om ni så önskar.

# 3 Uppgifter

## 3.1 Grunduppgift

Applicera Gauss-Newtons metod och Levenberg-Marquardts metod på ellipsproblemet. Använd $\epsilon = 10^{-3}$ och begränsa antalet iterationer till 100 för bägge metoderna.

Använd $\mu = 0.01, \alpha_{\text{min}} = 10^{-3}$ i Gauss-Newtons metod.

Använd $\Delta_0 = 1, \mu = 0.25, \eta = 0.75$ i Levenberg-Marquardts metod.

- Verifiera att metoderna konvergerar mot $x_*$ från rimligt stort område då $\sigma = 0$.
- Undersök hur metoderna uppför sig under realistiska förhållanden för olika nivåer på felet $\sigma = 10^k$, för $k = -3, -2, \ldots$. Lösningspunkter, antal iteration, konvergenshastighet, etc.
- Testa hur metoderna uppför sig med minimalt antal mätpunkter för olika stort fel.
- Testa hur metoderna uppför sig då $x_*$ motsvarar rangdefekt eller illa konditionerad lösning (cirkel eller utdragen ellips).
- Testa hur metoderna uppför sig då någon/några av mätpunkterna ger problem (t.ex. mätpunkt i $(c_x, c_y)$ eller på insidan, nära spets av utdragen ellips).

## 3.2 Extrauppgift 1

- Analysera hur valet av $\mu$ och $\alpha_{min}$ påverkar Gauss-Newtons metod. Konvergens, risk för krasch, stagnering, etc. Vad händer om talen väljs för högt/för lågt? Motivera och redovisa exempel i lämpliga fall.
- Analysera hur valet av $\Delta_0, \mu, \eta$ påverkar trust-region-metoden. Konvergens, risk för krasch, stagnering, etc. Vad händer om talen väljs för högt/för lågt? Motivera och redovisa exempel i lämpliga fall.

## 3.3 Extrauppgift 2

Välj något av följande:

1. Implementera en funktion som returnerar $Q(x)$, den kvadratiska delen av hessianen. Implementera dessutom en modifierad Gauss-Newton (egentligen Newton) som utnyttjar den fullständiga hessianen och den modifierade $LDL^T$-faktorisering (utdelas). Jämför konvergenshastigheterna för Gauss-Newton och modifierad Newton för olika storlekar på residualen.
2. Tag fram en väl fungerande bestämning av startvärdet $x_0$ baserad endast på mätvärdena, dvs som det skulle kunna vara i verkligheten. Demonstrera att det fungerar väl för en stor variation av problem.
3. Modifiera implementationen av problemet eller algoritmerna så att parametrarna i lösningen är av ungefär samma magnitud, dvs så att lösningsvektorn är väl skalad. Du får anta att magnituden hos lösningen är känd i förväg. Jämför lösningarna på det ursprungliga problemet och det skalade.

### 3.4 Realistiska data

Exempel på realistiska data utdelas separat.

## 4 Betyg

Korrekt löst grunduppgift ger betyget 3 på laborationen. Varje extrauppgift 1 grad till. Även kvalitén på lösningen och rapporten kan påverka betyget.

Förutom det som beskrivs ovan ska källkoden ingå i rapporten. Källkoden skall vara väl skriven, indenterad och kommenterad.

Uppgiften skall utföras enskilt och skall inlämnas senast kl. 13.30 den 6 december 1999.

Samtliga uppgifter skall vara kompletterade senast kl. 09.30 den 23 januari 2000.