# MMGF11 - Analys och linjär agebra, 2015

**Laborationsuppgift nummer I, del 3: Kryptografi - Hill's chiffer.**

Vi ska studera ett enkelt sätt att koda och avkoda hemliga meddelanden med hjälp av linjär algebra, närmare bestämt med linjära transformationer med en matris resp. lösning av ekvationssystem med samma matris. Vi ska även se på hur man kan avslöja den använda kodningsnyckeln dvs. den aktuella matrisen och på så sätt *knäcka koden*. Det okodade meddelandet kallar vi *klartext* och det kodade kallar vi *shiffer*.
En enkel kodningsnyckel är att ersätta varje bokstav i alfabetet med en annan, ett exempel på ett så kallat substitutionschiffer. Det är enkelt att knäcka en sådan kod eftersom man vet frekvenserna av olika bokstäver i ett språk.
Här gör vi nu så att vi grupperar bokstäverna i klartexten och kodar grupp för grupp i stället för bokstav för bokstav. Om antalet bokstäver i en grupp är $n$ kan vi ersätta dessa med en grupp av $n$ chiffertecken genom en linjär transformation dvs. genom matris-vektor multiplikation med en $n \times n$-matris. Vi betraktar alltså en grupp av $n$ tecken som en kolonnvektor, som efter transformationen blir en ny kolonnvektor med $n$ tecken. En förutsättning för att det ska fungera är att vi kan konvertera tecken till tal men det går bra i MATLAB. Funktionen *abs* konverterar från tecken till heltal och funktionen *char* konverterar tillbaka heltal till tecken. Med en heltalsmatris som kodningsnyckel fungerar alltså det hela. Man kan lämpligen använda rutinen *reshape* när man kodar hela texten på en gång.
Så här ser en kodningsfunktion i MATLAB ut för fallet $n = 4$. A är en given kodningsmatris av ordning $4 \times 4$, $p$ är klartexten och funktionens värde blir chiffret:

```
function f=encode(p,A)
p=[p,' ',' ',' ',' '];
m=floor(length(p)/4);
pp=abs(reshape(p(1:4*m),4,m));
cc=A*pp;
f=char(reshape(cc,1,4*m));
```

Avkodning innebär den *inversa proceduren* dvs. lösning av linjärt ekvationssystem med samma kodningsmatris. För att kunna läsa meddelandet krävs att man känner till kodningsnyckeln dva.s matrisen, som vi antar är ickesingulär (reguljär, inverterbar).

**Uppgifter:**
**a.** Låt $n = 4$. Skriv en funktionsfil i MATLAB för avkodning, gärna med inspiration från kodningsfunktionen ovan. Det är alltså viktigt att förstå koden ovan för att kunna klara av denna uppgift. Hur mycket behöver du egentligen ändra i den för att det ska bli avkodning i stället för kodning? Tolka meddelandet i variabeln *code_1* i filen *krypto.mat*, som kan hämtas från kursens hemsida. Den använda kodningsmatrisen är $A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$.

**b.** Om man känner till de $n^2$ första tecknen i klartexten (och motsvarande chiffer, förstås) så kan man avslöja kodningsmatrisen. Man bör alltså undvika traditionella hövliga inledningsfraser i dessa sammanhang. Vi antar att dessa inledande tecken är sådana att motsvarande vektorer blir linjärt oberoende. Man kan då bestämma matrisen $A$ genom att lösa ett *system med flera högerled* med någon av funktionerna / (slash) eller \ (backslash) i MATLAB. Låt nu $n = 4$. Skriv en funktionsfil i MATLAB som knäcker koden dvs. bestämmer kodningsmatrisen $A$ och använd sedan matrisen och din avkodningsfunktion för att läsa hela det krypterade meddelandet i variabeln *code_2* i filen *krypto.mat*. Meddelandet inleds med klartexten: *The significant problems.*

**Anmärkningar:**
**1.** Om du kodar själv. Begränsa dig till *små heltal* i matrisen.
**2.** Se upp med avrundningsfel. Du vet att kodningsmatrisen ska vara en heltalsmatris.

**Vid redovisning krävs.** Funktionsfiler och körningsresultat dvs. klartexterna och kodningsnycklarna (matriserna).