

**HAMat-studiet.**

Skriftlig hjemmeopgave i Numerisk Analyse og Datalogi.

Eksamen, maj 2020.

Opgaven indeholder: I alt 47 sider: 11 sider opgavetekst og 36 sider bilag.

Opgaven udleveres : Den 14.05.2020 på Canvas fra kl. 9.00.

Opgaven afleveres : Den 28.05.2020 elektronisk i Digital Eksamen, senest kl. 12.00.

Gruppestørrelse : 1 studerende: Eksamen er individuel.

Eksamensregler : Det fastsatte maksimale sidetal i den afleverede projektrapport er 40 sider, som ikke inkluderer bilag. F.eks skal **hele** den udviklede C++ kode placeres i et bilag. Projektrapport inkl. bilag afleveres i **ét** dokument.

Spørgetimer : I form af adgang til konsultation via mail/Skype i følgende tidsrum:

Den 14.05.2020 kl. 10.00 – 12.00

Den 15.05.2020 kl. 10.00 – 12.00

Den 18.05.2020 kl. 10.00 – 12.00

Den 20.05.2020 kl. 10.00 – 12.00

Den 22.05.2020 kl. 10.00 – 12.00

Den 25.05.2020 kl. 10.00 – 12.00

**De to hovedemner for denne opgave er I og II.**I: Bestemmelse af mindste kvadraters løsning  $\mathbf{x}^*$  til et overbestemt system af lineære ligninger. $\mathbf{Ax} = \mathbf{b}$  hvor  $\mathbf{A}(n \times m)$  er koefficientmatricen $\mathbf{b}(n \times 1)$  er vektoren af højresider $\mathbf{x}(m \times 1)$  er vektoren af ubekendte, og hvor $n > m$  dvs at systemet har flere ligninger end ubekendte.II: Bestemmelse af egenløsninger  $\{\lambda_k, \mathbf{v}_k\}$  med  $k = 1, 2, \dots, n$  til en symmetrisk matrix  $\mathbf{A}(n \times n)$ .På grund af sin symmetri har  $\mathbf{A}(n \times n)$   $n$  reelle egenverdier  $\lambda_k$  og  $n$  ortogonale egenvektorer  $\mathbf{v}_k$ .**Bestemmelsen af  $\mathbf{x}^*$**  skal ske efter tre forskellige metoder 1), 2) og 3):1)  $\mathbf{Ax} = \mathbf{b} \Rightarrow \mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}$  som er normalligningerne og løses ved Gauss elimination, delvis pivotering og backwards substitution.2)  $\mathbf{Ax} = \mathbf{b} \Rightarrow \mathbf{Rx} = \mathbf{Q}^T \mathbf{b}$  hvor **QR**-faktoriseringen  $\mathbf{A} = \mathbf{QR}$  er frembragt ved modificeret Gram-Schmidt ortogonalisering af  $\mathbf{A}$ . $\mathbf{Rx} = \mathbf{Q}^T \mathbf{b}$  løses ved backwards substitution og giver  $\mathbf{x}^*$ .3) Minimering af  $f(\mathbf{x})$  hvor  $f(\mathbf{x}) = \|\mathbf{b} - \mathbf{Ax}\|^2 = (\mathbf{b} - \mathbf{Ax})^T (\mathbf{b} - \mathbf{Ax})$  er en kvadratisk form og dermed en funktion af de  $m$  variable  $\mathbf{x}^T = [x_1 \ x_2 \ \dots \ x_m]$  og hvor:  $f(\mathbf{x}^*) = \min f(\mathbf{x})$ .Minimering af  $f(\mathbf{x})$  udføres ved en iterativ algoritme.**Bestemmelse af egenløsningerne  $\{\lambda_k, \mathbf{v}_k\}$**  for  $\mathbf{A}(n \times n)$  skal ske ved metoden 4), som har sit udgangspunkt i spektralfremstillingen af  $\mathbf{A}(n \times n)$ , hvor  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{n-1}, \mathbf{v}_n$  er **ortho-normale**:

$$\mathbf{A} = \mathbf{S} \cdot \mathbf{\Lambda} \cdot \mathbf{S}^{-1} = \mathbf{S} \cdot \mathbf{\Lambda} \cdot \mathbf{S}^T = >$$

$$\mathbf{A} = \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T + \lambda_3 \mathbf{v}_3 \mathbf{v}_3^T + \dots + \lambda_k \mathbf{v}_k \mathbf{v}_k^T + \dots + \lambda_{n-1} \mathbf{v}_{n-1} \mathbf{v}_{n-1}^T + \lambda_n \mathbf{v}_n \mathbf{v}_n^T$$

4) Potensmetoden anvendes på de  $n$  matricer  $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4, \dots, \mathbf{P}_n$  hvor

$$\mathbf{P}_1 = \mathbf{A},$$

$$\mathbf{P}_2 = \mathbf{A} - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T,$$

$$\mathbf{P}_3 = \mathbf{A} - (\lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T),$$

$$\mathbf{P}_4 = \mathbf{A} - (\lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T + \lambda_3 \mathbf{v}_3 \mathbf{v}_3^T),$$

:

$$\mathbf{P}_n = \mathbf{A} - (\lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T + \lambda_3 \mathbf{v}_3 \mathbf{v}_3^T + \dots + \lambda_{n-1} \mathbf{v}_{n-1} \mathbf{v}_{n-1}^T)$$

**Tilvejebringelse af datagrundlaget**  $n, m, \mathbf{A}(n \times m)$  og  $\mathbf{b}(n \times 1)$  skal ske på 4 forskellige måder:

- Indlæsning af  $n, m$  og samtlige elementer i  $\mathbf{A}$  og  $\mathbf{b}$  fra en datafil med prædefineret struktur.
- Brugerindtastning af  $n, m$  og samtlige elementer i  $\mathbf{A}$  og  $\mathbf{b}$ .
- Frembringelse af  $\mathbf{A}$  og  $\mathbf{b}$  ud fra et simuleret observationssæt  $(x_i, y_i, z_i)$  for en lineær, multipel regressionsmodel:  $z_i = a \cdot x_i + b \cdot y_i + c + \varepsilon_i$  med  $i = 1, 2, \dots, n$ . Koefficienterne  $a, b$  og  $c$  brugervalgte, og  $\varepsilon_i$  modellerer en ikke-kontrollerbar tilfældig/stokastisk afvigelse, her i form af simulerede, computergenererede observationer fra en normalfordeling. Benyttes c) gælder altid  $m = 3$ , mens  $n$  er brugervalgt, altså:  $\mathbf{A}(n \times 3)$ .
- Fremstilling af  $\mathbf{A}$  og  $\mathbf{b}$  ved konstruktion af overbestemte ligningssystemer  $\mathbf{Ax} = \mathbf{b}$  med brugervalgt mindste kvadraters løsning  $\mathbf{x}^*$  og mindste kvadraters fejl  $\mathbf{e} = \mathbf{b} - \mathbf{Ax}^*$ .

**Projektopgaven er opdelt i følgende 5 opgaver:**

- Opgave 1: Implementering, test og dokumentation af den givne grundstruktur samt a) og b).  
 Generering og fordelingskontrol af de tilfældige, normalfordelte  $e_i$  som indgår i c).  
 Opgave 2: Implementering, test og dokumentation af c) og 1).  
 Opgave 3: Implementering, test og dokumentation af 4).  
 Opgave 4: Implementering, test og dokumentation af d) og 2).  
 Opgave 5: Implementering, test og dokumentation af 3).

**Opgaveteksten ledsages af 36 sider bilag:**

- Side 1 – 4: Potensmetoden.  
 Side 4 – 6: Udvidelse af potensmetoden.  
 Side 7 – 12: QR-faktoriseringer, løsning af overbestemte ligningssystemer ved hjælp af QR-faktoriseringer.  
 Side 13 – 16: (Smal) QR-faktorisering ved modificeret Gram-Schmidt ortogonalisering.  
 Side 17 – 18: (Smal) QR-faktorisering ved traditionel Gram-Schmidt ortogonalisering.  
 Side 19 – 20: Brugerdefineret fremstilling af et overbestemt ligningssystem med kendt, selvvalgt mindste kvadraters løsning. Fremstillingen sker ud fra en QR-strategi.  
 Side 21 – 24: Algoritmer til minimering af en multivariabel funktion  $f(\mathbf{x})$ .  
 Side 25 – 27: Centrale begreber for multivariable funktioner  $f(\mathbf{x})$ , her med  $\mathbf{x}^T = [x_1 \ x_2]$ .  
 Side 28 – 30: Studium af tilnærmelsen til  $f(\mathbf{x})$  ved hjælp af en 2. ordens Taylorudvikling.  
 Side 31 – 36: Iterativ minimering af en testfunktion  $f(\mathbf{x})$  ved anvendelse af en algoritme fra samme klasse af algoritmer som den, som skal implementeres i opgave 5.

## C++

Skriv et C++ program til

- Bestemmelse af mindste kvadraters løsning til et overbestemt system af lineære ligninger.
- Bestemmelse af egenløsningerne til en symmetrisk matrix.

Programmet skal overholde følgende krav og specifikationer:

## Opgave 1

Udfør implementering, test og dokumentation af:

- den givne grundstruktur, som vises i rutediagrammet s. 11 i opgaveteksten.
- frembringelsen af datagrundlag som specificeret ved a) og b).
- Generering og fordelingskontrol af tilfældige, normalfordelte  $e_i$  hvor  $i = 1, 2, \dots, n$ .

**Først: Implementering og afprøvning af grundstrukturen.**

Programmet skal implementere den grundstruktur, der præsenteres på side 11 i opgaveteksten i det viste rutediagram. Bemærk, at hele strukturen skal implementeres, uanset hvor stor en del af den samlede hjemmeopgave man besvarer. Se side 44 og 45 i noterne, der viser et rutediagram og en tilhørende implementering af grundstrukturen .

Den færdige besvarelse af eksamensopgaven skal indeholde to selvstændige programmer. Ét, som **kun** implementerer grundstrukturen **uden** at udføre beregninger. Samt det endelige program, som er en videreudvikling af den tomme grundstruktur, suppleret med diverse funktioner. I opgave 1 drejer det (kun) sig om det program, der **kun** implementerer grundstrukturen, og dette program skal fremstilles efter følgende

**Kravsspecifikation:**

- Hele grundstrukturen på side 11 skal implementeres i main-funktionen.
- De **eneste** variable, der erklæres i main-funktionen, er 7 valgvariable svarende til de rombeformede ”valgkasser” i rutediagrammet:  
**char Valgabcd, ValgGemB, ValgGemD, ValgKontrolr, Valg124, Valg3, ValgIgen;**  
Brugervalg af indholdet af disse variable muliggør dermed afprøvning af alle forgreninger.
- Forud for ethvert valg/forgreningspunkt præsenteres de mulige alternativer for brugeren, som derefter træffer sit valg ved indtastning af en værdi til den aktuelle variabel. F.eks.:  
Forud for valget mellem a), b), c) og d) findes programlinierne:  
**cout<<”\nVælg a, b, c eller d her: Valgabcd =”;**  
**cin>>Valgabcd;**
- Efter realiseringen af et valg findes en **cout-sætning**, der udskriver en meddelelse om hvilket valg, der blev truffet. F.eks.:  
**cout<<”\nDu indtastede: b - dermed valgte du b): Indtastning af n, m, A og b”;**
- I det selvstændige program, der **kun** implementerer grundstrukturen, udføres **ingen** beregninger. Men enhver af de rektangulære ”handlingskasser” i rutediagrammet repræsenteres af en **cout-sætning**, som udskriver en ultra-kort beskrivelse af den pågældende handling. F.eks.:  
**cout<<”\nPå dette sted i programmet udføres: Indtastning af n, m, A og b”;**
- Dokumentér to tests af programmet, som kun implementerer grundstrukturen:  
En kørsel, hvor der i samtlige valg/forgreninger aktiveres den gren, som er længst til venstre. Og en kørsel, hvor der i samtlige valg/forgreninger aktiveres den gren, der er længst til højre.
- Den videre programudvikling sker med den hermed implementerede skabelon af grundstrukturen som grundlag og udgangspunkt.

**Tilvejebringelse af n, m, A og b:**

a) Indlæsning af **A** og **b** fra en datafil, som skal have følgende struktur:

```

n
m
a11 a12 a13 ..... a1m b1
a21 a22 a23 ..... a2m b2
.   .   .           .   .
.   .   .           .   .
an1 an2 an3 ..... anm bn

```

b) Brugerindtastning af n og m og samtlige elementer i **A** og **b** med mulighed for at rekvirere, at de indtastede data gemmes i en fil. Filen skal have den struktur, som er vist under a).

### Frembringelse af vektoren $\mathbf{r}$ ( $n \times 1$ )

Vektoren  $\mathbf{r}$ , hvor  $\mathbf{r}^T = [r_1 \ r_2 \ r_3 \ \dots \ r_n]$ , indeholder:

$n$  simulerede observationer  $r_i$   $i = 1, 2, \dots, n$  fra standardnormalfordelingen  $N(0, 1)$ .

Simulering af to observationer  $r_A$  og  $r_B$  fra normalfordelingen  $N(0, 1)$  udføres således:

- Brug C++ funktionerne `rand()` og `srand()` fra `stdlib.h` og dan  $x$ :  
 $x = (\text{rand}() \% 10001) / 10000.0$ ; //  $x$  er dermed et tilfældigt tal fra intervallet  $[0; 1]$
- Generér to tilfældige tal  $x_A$  og  $x_B$  fra intervallet  $[0; 1]$ .
- Beregn:  $r_A = \sqrt{(-2 \cdot \ln(x_A))} \cdot \cos(2\pi x_B)$  og  $r_B = \sqrt{(-2 \cdot \ln(x_A))} \cdot \sin(2\pi x_B)$ .

Med  $n = 100$  dannes vektoren  $\mathbf{r}$ , hvor  $\mathbf{r}^T = [r_1 \ r_2 \ r_3 \ \dots \ r_{100}]$  ved at generere 50 observationspar  $(r_A, r_B)$ .

Alternativ og godkendt fremgangsmåde til simulering af tilfældige tal fra intervallet  $[0; 1]$ :

```
#include <random> // Fra Canvas, modul 5, randomvektor.cpp
//Genererer en vektor med n koordinater.
// Koordinaterne er ligefordelte på [0,1] og uafhængige.
void randomvektor(double a[NMAX], int n)
{
    random_device rd;
    mt19937 gen(rd());
    uniform_real_distribution<double> dis(0.0, 1.0);
    for(int i = 0; i < n; i++) a[i]=dis(gen);
}
```

### Brugervalg af: Fordelingskontrol af de $n$ simulerede observationer i vektoren $\mathbf{r}$ ( $n \times 1$ ).

Hvis genereringen fungerer efter hensigten, vil  $r_1 \ r_2 \ r_3 \ \dots \ r_n$  være observationer fra standardnormalfordelingen  $N(0, 1)$ . I så fald vil hvert af de følgende 8 intervaller  $]-\infty; x_1]$  samt  $]x_k; x_{k+1}]$  med  $k = 1, 2, \dots, 6$  og  $]x_7; \infty[$  indeholde ca.  $1/8$  af observationerne  $r_1 \ r_2 \ r_3 \ \dots \ r_n$ . Se noter side 38 og 1. kursusopgave for definition af  $\Phi(x)$ .

Intervalgrænserne  $x_k$  med  $k = 1, 2, \dots, 7$  er givet således:  $\Phi(x_k) = 0.125 * k$ .

F.eks, hvor de viste værdier af  $x_0$  er startpunkter for en Newton-Raphson iteration:

- $\Phi(x_1) = 0.125 \Rightarrow x_1$  er løsning til problemet:  $\Phi(x) - 0.125 = 0 \Rightarrow f(x) = 0$ , hvor:  
 $x_1$  findes ved NR-iteration med:  $f(x) = \Phi(x) - 0.125 \Rightarrow f'(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$  og  $x_0 = -1.2$
- $\Phi(x_2) = 0.250 \Rightarrow x_2$  er løsning til problemet:  $\Phi(x) - 0.250 = 0 \Rightarrow f(x) = 0$ , hvor:  
 $x_2$  findes ved NR-iteration med:  $f(x) = \Phi(x) - 0.25 \Rightarrow f'(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$  og  $x_0 = -0.7$
- $\Phi(x_3) = 0.375 \Rightarrow x_3$  er løsning til problemet:  $\Phi(x) - 0.375 = 0 \Rightarrow f(x) = 0$ , hvor:  
 $x_3$  findes ved NR-iteration med:  $f(x) = \Phi(x) - 0.375 \Rightarrow f'(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$  og  $x_0 = -0.4$
- De øvrige intervalgrænser:  $x_4 = 0$ ,  $x_5 = -x_3$ ,  $x_6 = -x_2$  og  $x_7 = -x_1$ .

Derefter skal der foretages en optælling af antallet af observerede værdier af  $r_1 \ r_2 \ r_3 \ \dots \ r_n$  i hvert af de 8 intervaller. Brugeren vælger  $n$ . Med  $n = 80$  observationer, skal der altså ligge ca. 10 observationer i hvert af de 8 intervaller. Brugeren skal kunne vælge denne fordelingskontrol af  $\mathbf{r}$ .

Frembringelse af  $\mathbf{A}$  og  $\mathbf{b}$  ud fra et simuleret observationssæt  $(x_i, y_i, z_i)$  for en lineær, multipel regressionsmodel sker på grundlag af:  $z_i = a \cdot x_i + b \cdot y_i + c + \varepsilon_i$  med  $i = 1, 2, \dots, n$ .  
 $\varepsilon_i$  modellerer en ikke-kontrollerbar tilfældig afvigelse, som simuleres således:  $\boldsymbol{\varepsilon} = \sigma \cdot \mathbf{r}$ ,  
hvor  $\boldsymbol{\varepsilon}^T = [\varepsilon_1 \ \varepsilon_2 \ \varepsilon_3 \ \dots \ \varepsilon_n]$   $\Rightarrow \varepsilon_i$  er observationer fra normalfordelingen  $N(0, \sigma^2)$ .

## Opgave 2 ( Se D. Lay ed. 5 side 387)

1. Bestemmelsen af mindste kvadraters løsningen  $\mathbf{x}^*$  skal ske efter metoden 1):  
 $\mathbf{Ax} = \mathbf{b} \Rightarrow \mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}$  som er normalligningerne, og som løses ved Gauss elimination, delvis pivotering og backwards substitution.
  2. Frembringelse af datagrundlaget c).
- c) Frembringelse af  $\mathbf{A}$  og  $\mathbf{b}$  ud fra et simuleret observationssæt  $(x_i, y_i, z_i)$  for en lineær, multipel regressionsmodel:  $z_i = a \cdot x_i + b \cdot y_i + c + \varepsilon_i$  med  $i = 1, 2, \dots, n$ . Koefficienterne  $a, b$  og  $c$  brugervalgte, og  $\varepsilon_i$  modellerer en ikke-kontrollerbar tilfældig/stokastisk afvigelse, her i form af simulerede, computergenererede observationer fra en normalfordeling  $N(0, \sigma^2)$ . Benyttes c) gælder altid  $m = 3$ , mens  $n$  er brugervalgt, altså:  $\mathbf{A}(n \times 3)$ .

Strategien for frembringelsen af datagrundlaget c) tager udgangspunkt i regressionsmodellen.

**Den multiple regressionsmodel**  $z_i = a \cdot x_i + b \cdot y_i + c + \varepsilon_i$  med  $i = 1, 2, \dots, n$  udtrykker:

- Der findes en lineær sammenhæng mellem de to fri variable  $(x, y)$  og middelværdien  $E(Z)$  af den afhængige variabel  $Z$ .  $Z$  er en stokastisk variabel og  $z$  er en observation af  $Z$ .
- Værdien af  $z = a \cdot x + b \cdot y + c + \varepsilon$  er sammensat af to bidrag.  
 Det ene bidrag er kontrollerbart og ikke-stokastisk:  
 $a \cdot x + b \cdot y + c$  som udtrykker at værdierne  $(x, y)$  har en lineær påvirkning på  $z$ .  
 Det andet bidrag er ikke-kontrollerbart:  $\varepsilon$ , som må modelleres og analyseres statistisk.
- I praksis udtrykker  $z = a \cdot x + b \cdot y + c + \varepsilon$ , at der **findes** en lineær påvirkning  $a \cdot x + b \cdot y + c$ , som **delvis forklarer**  $z$ . Men regressionskoefficienterne  $a, b$  og  $c$  er ukendte og estimeres ud fra et datagrundlag af målinger  $(x_i, y_i, z_i)$  med  $i = 1, 2, \dots, n$ .

**Simuleringsstrategi.** Frembring datagrundlaget  $(x_i, y_i, z_i)$  med  $i = 1, 2, \dots, n$  således:

- Vælg  $x$ -værdier:  $x_k$  for  $k = 1, 2, \dots, m_x$  som udgør mængden  $S_x = \{x \mid x_1, x_2, \dots, x_{m_x}\}$
- Vælg  $y$ -værdier:  $y_k$  for  $k = 1, 2, \dots, n_y$  som udgør mængden  $S_y = \{y \mid y_1, y_2, \dots, y_{n_y}\}$
- Frembring et "gitter" af målepunkter:  $(x_i, y_i)$  med  $i = 1, 2, \dots, n$  hvor  $n = m_x \cdot n_y$ .  
 Altså:  $(x_i, y_i)$  med  $i = 1, 2, \dots, n$  er elementerne i mængdeproduktet  $S_{xy} = S_x \times S_y$ .
- Vælg regressionskoefficienterne  $a, b$  og  $c$ .
- Vælg standardafvigelsen  $\sigma > 0$  og **simulér**  $\boldsymbol{\varepsilon} = \sigma \cdot \mathbf{r}$  hvor  $\boldsymbol{\varepsilon}^T = [\varepsilon_1 \ \varepsilon_2 \ \varepsilon_3 \ \dots \ \varepsilon_n]$   
**Bemærk**, at  $\sigma$  regulerer et "støjniveau": Hvis  $\sigma$  er stor, vil det tilfældige bidrag  $\varepsilon$  til  $z$  kunne overdøve det kontrollerbare bidrag  $a \cdot x + b \cdot y + c$ , så det ikke kan registreres.
- Beregn de **simulerede** målinger af  $z_i = a \cdot x_i + b \cdot y_i + c + \varepsilon_i$  med  $i = 1, 2, \dots, n$ .
- Dan  $\mathbf{A}(n \times 3)$  og  $\mathbf{b}(n \times 1)$  og dermed det overbestemte ligningssystem  $\mathbf{Ax} = \mathbf{b}$ . Find: Mindste kvadraters løsningen  $\mathbf{x}^* = [a^* \ b^* \ c^*]$  af estimerede regressionskoefficienter.

**Implementerbar præsentation af simuleringsstrategien** med efterfølgende eksempel.

- Frembring  $x_k$  for  $k = 1, 2, \dots, m_x$  således:  
 Vælg  $x_1$ ,  $\Delta x$  og  $m_x$  og dan  $x_k = x_{k-1} + \Delta x$  for  $k = 2, 3, \dots, m_x$   
 Dan vektoren  $\mathbf{x}$  hvor  $\mathbf{x}^T = [x_1 \ x_2 \ x_3 \ \dots \ x_{m_x}]$ .
- Frembring  $y_k$  for  $k = 1, 2, \dots, n_y$  således:  
 Vælg  $y_1$ ,  $\Delta y$  og  $n_y$  og dan  $y_k = y_{k-1} + \Delta y$  for  $k = 2, 3, \dots, n_y$   
 Dan vektoren  $\mathbf{y}$  hvor  $\mathbf{y}^T = [y_1 \ y_2 \ y_3 \ \dots \ y_{n_y}]$ .
- Vælg regressionskoefficienterne  $a, b$  og  $c$  samt standardafvigelsen  $\sigma$ .
- Dan vektoren  $\mathbf{e}_{m_x}(m_x \times 1)$  hvis  $m_x$  koordinater **alle** er tallet 1. Altså  $m_x$  et-taller.  
 Dan vektoren  $\mathbf{e}_{n_y}(n_y \times 1)$  hvis  $n_y$  koordinater **alle** er tallet 1. Altså  $n_y$  et-taller.
- Dan matrix  $\mathbf{X} = \mathbf{e}_{n_y} \cdot \mathbf{x}^T \Rightarrow \mathbf{X}(n_y \times m_x)$  indeholder alle  $x$ -koordinaterne til de  $n$  målepunkter  $(x_i, y_i)$  med  $i = 1, 2, \dots, n$ .

- Dan matrix  $\mathbf{Y} = \mathbf{y} \cdot \mathbf{e}_{mx}^T \Rightarrow \mathbf{Y}(n_y \times m_x)$  indeholder alle y-koordinaterne til de  $n$  målepunkter  $(x_i, y_i)$  med  $i = 1, 2, \dots, n$ .
- Dan vektorerne  $\mathbf{v}_x$  og  $\mathbf{v}_y$  ved stabling af søjlerne i henholdsvis  $\mathbf{X}$  og  $\mathbf{Y}$ .
- Dan  $\mathbf{A}(n \times 3)$  således:  $\mathbf{v}_x$  og  $\mathbf{v}_y$  indsættes som henholdsvis 1. og 2. søjle i  $\mathbf{A}(n \times 3)$ . Den 3. søjle i  $\mathbf{A}(n \times 3)$  har tallet 1 på alle  $n$  pladser. Altså en søjle med  $n$  et-taller.
- Beregn vektor  $\mathbf{z}_0 = \mathbf{A} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix}$
- Beregn vektor  $\boldsymbol{\varepsilon} = \sigma \cdot \mathbf{r}$ , hvor  $\mathbf{r}^T = [r_1 \ r_2 \ \dots \ r_n]$  fås ved simulationer af  $N(0,1)$ .
- Beregn  $\mathbf{b} = \mathbf{z}_0 + \boldsymbol{\varepsilon}$ .
- Dermed er det overbestemte ligningssystem  $\mathbf{Ax} = \mathbf{b}$  defineret.

### Eksempel


Med  $x_1 = 1$   $\Delta x = 1$   $m_x = 4$  fås  $\mathbf{x}^T = [1 \ 2 \ 3 \ 4]$  og  $\mathbf{e}_{mx}^T = [1 \ 1 \ 1 \ 1]$

Med  $y_1 = 0$   $\Delta y = 1$   $n_y = 5$  fås  $\mathbf{y}^T = [0 \ 1 \ 2 \ 3 \ 4]$  og  $\mathbf{e}_{ny}^T = [1 \ 1 \ 1 \ 1 \ 1]$

$$\mathbf{X} = \mathbf{e}_{ny} \cdot \mathbf{x}^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \cdot [1 \ 2 \ 3 \ 4] = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{bmatrix} \Rightarrow$$

$\mathbf{v}_x^T = [1 \ 1 \ 1 \ 1 \ 1 \ 2 \ 2 \ 2 \ 2 \ 2 \ 3 \ 3 \ 3 \ 3 \ 3 \ 4 \ 4 \ 4 \ 4 \ 4]$  indeholder x-koordinater til  $(x_i, y_i)$

$$\mathbf{Y} = \mathbf{y} \cdot \mathbf{e}_{mx}^T = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \cdot [1 \ 1 \ 1 \ 1] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix} \Rightarrow$$

  $\mathbf{v}_y^T = [0 \ 1 \ 2 \ 3 \ 4 \ 0 \ 1 \ 2 \ 3 \ 4 \ 0 \ 1 \ 2 \ 3 \ 4 \ 0 \ 1 \ 2 \ 3 \ 4]$  indeholder y-koordinater til  $(x_i, y_i)$

$$\Rightarrow \mathbf{A}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 \\ 0 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Koordinater i  $\mathbf{b}(20 \times 1)$ :  $b_i = z_i = a \cdot x_i + b \cdot y_i + c + \sigma \cdot r_i$  for  $i = 1, 2, 3, \dots, 20$ .

### C++ kode til 2 relevante void-funktioner:

- Det ydre vektorprodukt  $\mathbf{a} \cdot \mathbf{b}^T$  af  $\mathbf{a}(n \times 1)$  og  $\mathbf{b}(m \times 1)$ :  $\mathbf{M} = \mathbf{a} \cdot \mathbf{b}^T$  hvor  $\mathbf{M}(n \times m)$ .
- Ovenfor:  $\mathbf{X} = \mathbf{e}_{ny} \cdot \mathbf{x}^T$  og  $\mathbf{Y} = \mathbf{y} \cdot \mathbf{e}_{mx}^T$

void YdreProd(double M[10][10], double a[10], double b[10], int n, int m)

```
{
    int i, j;
    for (i = 0; i < n; i = i + 1)
    {
        for (j = 0; j < m; j = j + 1)
        {
            M[i][j] = a[i] * b[j];
        }
    }
}
```

- Stabling af søjlerne i matrix.
- Ovenfor: Dan vektorerne  $\mathbf{v}_x$  og  $\mathbf{v}_y$  ved stabling af søjlerne i henholdsvis  $\mathbf{X}$  og  $\mathbf{Y}$ .

```
void StakCols( double M[10][10], double v[100], int n, int m)
{
    int i, j, a;
    for (j = 0; j < m ; j = j+1)
    {
        a = j * n;
        for (i = 0; i < n ; i = i + 1)
        {
            v[a + i] = M[i][j] ;
        }
    }
}
```

### Opgave 3

1. Implementér, test og dokumentér følgende algoritme:

**Algoritme** for den udvidede potensmetode til bestemmelse af  $\{\lambda_k, \mathbf{v}_k\}$   $k = 1, 2, 3, \dots, n$ .  
 Initialisering af iterationen  $k = 1, 2, 3, \dots, n$  til bestemmelse af  $\{\lambda_k, \mathbf{v}_k\}$

Vælg:  $\varepsilon$  tolerancen for potensmetodens stopkriterium.  
 N den øvre grænse for antallet af tilladte iterationer. (Nødbremse!)  
 Sæt:  $k = 1$  iterationsindeks (dvs egenløsnings-indeks) initialiseres  
 Sæt:  $\mathbf{P}_1 = \mathbf{A}$

Iteration  $k = 1, 2, 3, \dots, n$

Givet:  $\mathbf{P}_k$

Initialisering af iterationen  $i = 1, 2, 3, \dots$  til potensmetode-bestemmelse af  $\{\lambda_k, \mathbf{w}_k\}$

Vælg:  $\mathbf{z}_0$  startvektoren for iterationen.  
 Sæt:  $i = 0$  potensmetodens iterationstæller 0-stilles.  
 Find:  $L_0$  den numerisk største koordinat i  $\mathbf{z}_0$   
 Beregn:  $\mathbf{y}_0 = \frac{1}{L_0} \mathbf{z}_0$

Iteration  $i = 1, 2, 3, \dots$

Givet:  $\mathbf{y}_{i-1}$   
 Beregn:  $\mathbf{z}_i = \mathbf{P}_k \cdot \mathbf{y}_{i-1}$   
 Find:  $L_i$  den numerisk største koordinat i  $\mathbf{z}_i$   
 Beregn:  $\mathbf{y}_i = \frac{1}{L_i} \mathbf{z}_i$

Stopkriterium

Stop hvis:  $|L_i - L_{i-1}| < \varepsilon \quad \vee \quad |\mathbf{y}_i - \mathbf{y}_{i-1}| < \varepsilon \quad \vee \quad i = N$

Løsningskriterium

Ved terminering:  $i < N \Rightarrow \mathbf{y}_i \approx \mathbf{w}_k \quad \wedge \quad L_i \approx \lambda_k$

Gem:  $\mathbf{v}_k = \frac{\mathbf{w}_k}{|\mathbf{w}_k|}$  som søjle nr.  $k$  i matricen  $\mathbf{S}$  (jvf  $\mathbf{A} = \mathbf{S} \cdot \mathbf{\Lambda} \cdot \mathbf{S}^{-1} = \mathbf{S} \cdot \mathbf{\Lambda} \cdot \mathbf{S}^T$ ).  
 $\lambda_k$  som element nr.  $k$  i vektoren  $\mathbf{d}$  som er diagonalen i  $\mathbf{\Lambda}$ .

Opdater:  $\mathbf{P}_{k+1} = \mathbf{P}_k - \lambda_k \frac{1}{\mathbf{w}_k^T \mathbf{w}_k} \mathbf{w}_k \mathbf{w}_k^T$  (så iterationen mod  $\{\lambda_{k+1}, \mathbf{w}_{k+1}\}$  kan gå i gang)

Algoritmen leverer grundlaget for implementering af metode 4), som - forudsat at  $\mathbf{A}(n \times n)$  er symmetrisk med distinkte egenverdier – vil udføre en:

**Bestemmelse af egenløsningerne**  $\{\lambda_k, \mathbf{v}_k\}$  for  $\mathbf{A}(n \times n)$  ved følgende metode, som har sit udgangspunkt i spektralfremstillingen af  $\mathbf{A}(n \times n)$ , hvor  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{n-1}, \mathbf{v}_n$  er *ortho-normale*.

$$\mathbf{A} = \mathbf{S} \cdot \mathbf{\Lambda} \cdot \mathbf{S}^{-1} = \mathbf{S} \cdot \mathbf{\Lambda} \cdot \mathbf{S}^T = >$$

$$\mathbf{A} = \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T + \lambda_3 \mathbf{v}_3 \mathbf{v}_3^T + \dots + \lambda_k \mathbf{v}_k \mathbf{v}_k^T + \dots + \lambda_{n-1} \mathbf{v}_{n-1} \mathbf{v}_{n-1}^T + \lambda_n \mathbf{v}_n \mathbf{v}_n^T$$

Potensmetoden anvendes på de  $n$  matricer  $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4, \dots, \mathbf{P}_n$  hvor

$$\mathbf{P}_1 = \mathbf{A},$$

$$\mathbf{P}_2 = \mathbf{A} - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T,$$

$$\mathbf{P}_3 = \mathbf{A} - (\lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T),$$

$$\mathbf{P}_4 = \mathbf{A} - (\lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T + \lambda_3 \mathbf{v}_3 \mathbf{v}_3^T),$$

⋮

$$\mathbf{P}_n = \mathbf{A} - (\lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T + \lambda_3 \mathbf{v}_3 \mathbf{v}_3^T + \dots + \lambda_{n-1} \mathbf{v}_{n-1} \mathbf{v}_{n-1}^T)$$

- Eksterne tests skal udføres på testmatricer, som er konstrueret med kendte egenløsninger.
- Find egenløsninger for  $\mathbf{A}^T \mathbf{A}$  som er symmetrisk, positiv semidefinit og relevant ved bestemmelsen af mindste kvadraters løsningen for overbestemte systemer  $\mathbf{A} \mathbf{x} = \mathbf{b}$ .

#### Opgave 4

Implementering, test og dokumentation af

1. Frembringelse af datagrundlaget d): Se bilag side 19 – 20
2. Metode 2): Se bilag side 8 – 12, som beskriver QR-faktoriserings generelt.  
Se bilag side 13 – 16, som præsenterer den modificerede Gram-Schmidt ortogonalisering og dens tilhørende QR –faktoriserings implementering.

Bestemmelsen af mindste kvadraters løsningen  $\mathbf{x}^*$  sker efter metode 2) således:

$\mathbf{A} \mathbf{x} = \mathbf{b} \Rightarrow \mathbf{R} \mathbf{x} = \mathbf{Q}^T \mathbf{b}$  hvor QR-faktoriserings:  $\mathbf{A} = \mathbf{Q} \mathbf{R}$  er frembragt ved modificeret Gram-Schmidt ortogonalisering af  $\mathbf{A}$ .

$\mathbf{R} \mathbf{x} = \mathbf{Q}^T \mathbf{b}$  løses ved backwards substitution og giver  $\mathbf{x}^*$ .

Den beregnede mindste kvadraters løsning  $\mathbf{x}^*$  ønskes kontrolleret.

Matricerne  $\mathbf{A}_n$  og  $\mathbf{S}_n$  hvor  $n = 4, 8, 16$  præsenteres på side 19 – 20 i bilaget.

**Bemærk:** I dette projekt kræves kun, at brugeren kan vælge  $n = 4$  eller  $n = 8$ .

Bemærk:  $\mathbf{A}_4$  er en del af  $\mathbf{A}_8$ . Derfor kræves nedenfor at,  $\mathbf{A}_8$  findes i en datafil. Hvis brugeren vælger c) skal programmet indlæse  $\mathbf{A}_8$  fra denne fil.

Hvis brugeren vælger  $n = 4$  dannes  $\mathbf{A}_4$  som (4x4)–sektionen i øverste venstre hjørne af  $\mathbf{A}_8$ .

Sammenfattende tilvejebringes  $n, m, \mathbf{A}$  og  $\mathbf{b}$  således (se bilag side 19 – 20) :

- Indlæs  $\mathbf{A}_8$  fra en datafil.
- Vælg  $n = 4$  eller  $n = 8$  og dan  $\mathbf{A}_n$
- Indtast  $m$  (hvor  $m < n$ )
- Dan  $\mathbf{Q}$
- Indtast  $\mathbf{R}$
- Beregn  $\mathbf{A} = \mathbf{Q} \mathbf{R}$
- Indtast  $\mathbf{x}^*$  og beregn  $\mathbf{b}_p = \mathbf{A} \mathbf{x}^*$
- Indtast  $\mathbf{t}^*$  og beregn  $\mathbf{e}$
- Beregn  $\mathbf{b} = \mathbf{b}_p + \mathbf{e}$

Formålet med d) er at fremstille testeksempler med kendt  $\mathbf{x}^*$  og  $\mathbf{e}$ .



Derfor skal man have mulighed for at udskrive  $n$ ,  $m$ ,  $\mathbf{A}$  og  $\mathbf{b}$  til en fil med samme format som beskrevet under a), således at fremstillede testeksempler kan anvendes i forbindelse med afprøvning af løsningsmetoderne 1), 2) og 3).



Rekvireret designkontrol af c):

Vis og implementér beregning af  $\mathbf{e}$  ved projektion af  $\mathbf{b}$  på  $\text{Null}\{\mathbf{A}^T\}$ .

## Opgave 5

1. Implementering, test og dokumentation af metode 3). Se bilag side 21 – 24.

Bestemmelsen af mindste kvadraters løsningen  $\mathbf{x}^*$  til det overbestemte ligningssystem  $\mathbf{Ax} = \mathbf{b}$  skal ske ved minimering af den kvadratiske form:

$$f(\mathbf{x}) = \|\mathbf{b} - \mathbf{Ax}\|^2 = (\mathbf{b} - \mathbf{Ax})^T(\mathbf{b} - \mathbf{Ax}) \quad \text{hvor} \quad f(\mathbf{x}^*) = \min f(\mathbf{x})$$

Minimeringen skal udføres af en algoritme, der:

- Bestemmer søgeretningsvektorer  $\mathbf{s}_k$  ved Newton's metode:

$$\mathbf{v}_k = -[\nabla^2 f(\mathbf{x}_k)]^{-1} \cdot \nabla f(\mathbf{x}_k) \quad \Rightarrow \quad \mathbf{s}_k = \frac{1}{|\mathbf{v}_k|} \cdot \mathbf{v}_k$$

- Beregner gradienten således:  $\nabla f(\mathbf{x}) = 2\mathbf{A}^T\mathbf{Ax} - 2\mathbf{A}^T\mathbf{b}$
- Foretager en-dimensional minimering ved intervalreduktion udført ved Golden Section efter forudgående indkredsning af et startinterval for Golden Section.

Brugervalgte inddata:

$\mathbf{x}_0$	Startpunkt for algoritmen.
$\varepsilon_1$	Tolerance i stopkriteriet for algoritmen.
$N$	Det maksimalt tilladte antal iterationer for algoritmen.
$d$	Startsteplængden for indkredsningialgoritmen, som finder et startinterval for Golden Section.
$\varepsilon_2$	Tolerance i stopkriteriet for Golden Section.

$$f(\mathbf{x}) = (\mathbf{b} - \mathbf{Ax})^T(\mathbf{b} - \mathbf{Ax}) = \mathbf{b}^T\mathbf{b} - 2\mathbf{b}^T\mathbf{Ax} + \mathbf{x}^T\mathbf{A}^T\mathbf{Ax} \quad \Rightarrow \quad \nabla f(\mathbf{x}) = 2\mathbf{A}^T\mathbf{Ax} - 2\mathbf{A}^T\mathbf{b} \quad \Rightarrow$$

$$\text{Hessematrixen for } f(\mathbf{x}) \text{ er: } \nabla^2 f(\mathbf{x}) = 2\mathbf{A}^T\mathbf{A} \quad \Rightarrow \quad [\nabla^2 f(\mathbf{x})]^{-1} = \frac{1}{2} \cdot (\mathbf{A}^T\mathbf{A})^{-1}$$

Programmet skal beregne den inverse Hessematrix  $[\nabla^2 f(\mathbf{x})]^{-1}$  ud fra egenløsningerne for  $\mathbf{A}^T\mathbf{A}$ .

2. Fremstilling af datagrundlaget for et plot af  $(t, f(\mathbf{x}_0 + t \cdot \mathbf{s}_0))$  i intervallet  $0 \leq t \leq t_L$ .  
 Brugeren vælger  $t_L$ .  
 Underforstået her: Brugeren vælger også  $\mathbf{x}_0$ , som er startpunktet for algoritmen og dermed er blandt de brugervalgte inddata.  
 Det fremstillede datagrundlag skal udskrives til en datafil. Indholdet af denne datafil skal efterfølgende importeres i Mathematica, hvor det ønskede plot skal fremstilles.

**Formålet** med det fremstillede plot er at demonstrere: For ethvert valg af  $\mathbf{x}_0$  vil  $\mathbf{x}^*$  i *teorien* ligge på den rette linie  $\mathbf{x}(t) = \mathbf{x}_0 + t \cdot \mathbf{s}_0$ . Søgeretningsvektoren  $\mathbf{s}_0$  dannes som vist ovenfor. Dermed vil det eksisterende minimum i  $(\mathbf{x}^*, f(\mathbf{x}^*))$  kunne aflæses på plottet, *hvis*  $t_L$  er valgt passende og  $[\nabla^2 f(\mathbf{x})]^{-1}$  er beregnet præcist. Vis og diskutér, om formålet bliver indfriet.

### Note:

Side 99 i noterne orienterer om import af data til Mathematica.

Plottet kan laves ved hjælp af Mathematica-funktionen ListPlot[ ].

Det givne strukturoplæg til C++ programmet på side 11 i opgaveteksten skal implementeres. Al yderligere planlægning, design og udvikling af programmet overlades til programmørerne. Der vil ved bedømmelsen blive lagt vægt på, at det fremstillede program er velstruktureret og baseret på velvalgte functions, der overfører data via fornuftigt valgte parametre.

Der vil ligeledes blive lagt vægt på, at de eksterne tests er motiverede, samt udført og dokumenteret logisk og systematisk.

De interne tests ønskes dokumenteret for et passende udvalg af de vigtigste og mest centrale functions i programmet.

### **Mathematica**

Brug Mathematica systematisk til kontrolberegninger ved udførelsen af interne tests.

### **Opgavesættet slut**

#### **Overordnede retningslinier for karaktergivningen**

- 02 gives, når opgave 1 er løst fuldstændigt – d.v.s. både med hensyn til implementering, tests, dokumentation og relaterede aktiviteter i Mathematica.
- 4 gives, når yderligere opgave 2 er løst fuldstændigt.
- 7 gives, når yderligere opgave 3 er løst fuldstændigt.
- 10 gives, når yderligere opgave 4 er løst fuldstændigt.
- 12 gives, når yderligere opgave 5 er løst fuldstændigt .

Karakterskalaen bruges lineært og additivt med følgende karakterpoints for de 5 opgaver:

- 5 points for opgave 1 svarende til springet – 03 – 02.
- 2 points for opgave 2 svarende til springet 02 – 4.
- 3 points for opgave 3 svarende til springet 4 – 7.
- 3 points for opgave 4 svarende til springet 7 – 10.
- 2 points for opgave 5 svarende til springet 10 – 12.

En besvarelse, der vurderes til 80% af opgave 1, 50% af opgave 2, 100% af opgave 3, 0% af opgave 4 og 50% af opgave 5 krediteres med  $4 + 1 + 3 + 0 + 1 = 9$  karakterpoints. Det svarer til 6 på en talskala, der begynder med tallet – 03, nemlig  $9 - 03 = 6$ . Dermed gives 7 efter afrunding til nærmeste karakter.

# Brugerorientering om

Bestem mindste kvadraters løsningen (MKL)  $\underline{x}^*$  til  
det overbestemte ligningssystem (OLS)  $\underline{A}\underline{x} = \underline{b}$   
hvor  $\underline{A}(n \times m)$  og  $\underline{b}(n \times 1)$  med  $n > m$ :

11

- 1) Find  $\underline{x}^*$  som løsning til normalligningerne (NL):  $\underline{A}^T \underline{A} \underline{x} = \underline{A}^T \underline{b}$
  - 2) Udfør QR-faktorisering af  $\underline{A}$  ved modificeret G-S  
Find  $\underline{x}^*$  som løsning til:  $\underline{R} \underline{x} = \underline{Q}^T \underline{b}$
  - 3) Find  $\underline{x}^*$  ved minimering af:  $f(\underline{x}) = |\underline{b} - \underline{A}\underline{x}|^2$
  - 4) Find egenløsningerne  $\{\lambda_k, \underline{v}_k\}$   $k=1, 2, \dots, n$  for  $\underline{A}(n \times n)$  hvor  $\underline{A} = \underline{A}^T$
- Frembringelse af datagrundlag a), b), c) og d)
- a) Indlæs  $n, m, \underline{A}$  og  $\underline{b}$  fra en datafil
  - b) Indtast  $n, m, \underline{A}$  og  $\underline{b}$
  - c) Simuler målinger  $(x_i, y_i, z_i)$   $i=1, 2, \dots, n$  for en multipel regression  
 $z_i = ax_i + by_i + c + \varepsilon_i$  hvor  $\varepsilon_i$  er simulerede observationer af  $N(0, \sigma^2)$
  - d) Dan  $\underline{A}$  og  $\underline{b}$  så  $\underline{A}\underline{x} = \underline{b}$  har en brugervalgt  $\underline{x}^*$

