25. februar 2015/hhs/pch/pr

02601 Introduktion til numeriske algoritmer — Forår 2015

2. obligatoriske aflevering

Besvarelsen skal uploades som **pdf-fil** på CampusNet senest
onsdag den 8. april klokken 18:00.

Denne opgave drejer sig om behandling af fejlbehæftede signaler fra en stregkodelæser (kendt fra fx supermarkeder). Specifikt skal vi lave en algoritme, som kompenserer for at stregkodelæseren laver et uskarpt billede når den måler stregkoden.

I arbejdet med at løse denne opgave skal vi, i lighed med den første obligatoriske rapport, også træne nogle vigtige teknikker og beregningsmetoder fra *numerisk analyse*. Besvarelsen af disse spørgsmål træner de teknikker og metoder, som skal bruges til at designe og og analysere vores algoritmer – vi hjælper med disse træningsopgaver, mens opgaverne knyttet til algoritme-design skal løses selvstændigt.

- Spørgsmål 1–7 og I–IV skal besvares, inden man kan lave opgave **A** og **C**.
- Spørgsmål 8–9 og V–VI skal besvares, inden man kan lave opgave **B**.

Du skal besvare spørgsmål 1–9 og teori-spørgsmål I–VI og **C** (som omhandler de numeriske metoder) samt opgave **A** og **B** (der omhandler stregkodelæseren). Vedlæg Matlab-kode til alle relevante spørgsmål/opgaver i appendix.

Spørgsmål 1. Optælling af flops

En *floating-point operation* eller **flop** er en beregning (+, −, ∗ eller /) der involverer mindst et reelt tal. Eksempler: lad `i = 3`, `x = 5.6` og `y = pi/4`:

beregning	flops?	beregning	flops?
`i = i + 1`	nej	`j = 2*i + 1`	nej
`z = x + y`	1 flop	`x = x/3`	1 flop
`z = x - i`	1 flop	`z = 2*x + 2*y`	3 flops
`z = y^2`	1 flop	`z = x^i`	`i`−1 flops

Vi skal træne hvordan vi tæller flops i simple beregninger som involverer løkker.

1.1) Givet en vektor `x = [1.2 4.6 2.3 5 7.9 pi]`. Vi beregner summen af elementerne med denne løkke:

```
s = 0;
for i=1:length(x)
    s = s + x(i);
end
```

Hvor mange flops bruger denne løkke til at beregne summen `s`?

1.2) Givet en vektor `x` af typen double med `n` elementer. Vi beregner gennemsnit `g` og "root mean square" `RMS` med denne Matlab-kode:

```
g = 0; RMS = 0;
n = length(x);           % n = antal elementer i x
for i=1:n
    g = g + x(i);
    RMS = RMS + x(i)^2
end
g = g/n;
RMS = sqrt(RMS/n);
```

Hvor mange flops bruges til at beregne `g` og `RMS` (beregning af kvadratroden tæller som én flop)? Svaret skal udtrykkes ved `n`.

1.3) Algoritmen nedenfor tager en $\texttt{n} \times \texttt{n}$ tridiagonal matrix `A` og en højreside `b`, og foretager Gauss-elimination på matricen og reducerer højresiden tilsvarende:

```
for i = 1 : n-1
    factor = A(i+1,i) / A(i,i);
    A(i+1,i) = 0;
    A(i+1,i+1) = A(i+1,i+1) - factor*A(i,i+1);
    b(i+1) = b(i+1) - factor*b(i);
end
```

For eksempel, hvis input er

$$\texttt{A} = \begin{bmatrix} 2.2 & 1.1 & 0 & 0 & 0 \\ 3.3 & 4.4 & 5.5 & 0 & 0 \\ 0 & 6.6 & 7.7 & 8.8 & 0 \\ 0 & 0 & 1.5 & 2.6 & 3.7 \\ 0 & 0 & 0 & 1.8 & 2.9 \end{bmatrix}, \qquad \texttt{b} = \left\{\begin{matrix} 5.1 \\ 4.2 \\ 3.3 \\ 2.2 \\ 1.7 \end{matrix}\right\} \tag{1}$$

så bliver matrix og højreside ændret til

$$\texttt{A} = \begin{bmatrix} 2.2 & 1.1 & 0 & 0 & 0 \\ 0 & 2.75 & 5.5 & 0 & 0 \\ 0 & 0 & -5.5 & 8.8 & 0 \\ 0 & 0 & 0 & 5.0 & 3.7 \\ 0 & 0 & 0 & 0 & 1.5680 \end{bmatrix}, \qquad \texttt{b} = \left\{\begin{matrix} 5.1 \\ -3.45 \\ 11.58 \\ 5.3582 \\ -0.2289 \end{matrix}\right\}. \tag{2}$$

Hvor mange flops bruger algoritmen? Svaret skal udtrykkes ved `n`, og det skal forklares hvordan man kommer frem til udtrykket.

Spørgsmål 2. Skruer og møtrikker

Hér skal vi træne opstilling og løsning af et lineært ligningssystem.

2.1) Anders har 3 skruer, 12 møtrikker og 10 søm som vejer ialt 72.3 g. Birger har 12 skruer og 20 søm som vejer ialt 99.5 g. Carl har 2 møtrikker og 30 søm som vejer ialt 56.6 g. Vi antager at alle skruer vejer præcis det samme, og tilsvarende for møtrikker og søm. Opstil et 3×3 lineært ligningssystem $A\,x = b$ til beregning af vægten af en skrue, en møtrik og et søm.

2.2) Beregn løsningen til ligningssystemer ved hjælp af Matlabs "backslash" \ og angiv svaret med 2 decimaler.

2.3) Vi prøver nu at veje igen men med en anden vægt som viser forkert, og vi får derfor lidt forskellige resultater, nemlig 73.3 g, 98.4 g og 57.1 g for hhv Anders, Birger og Carl. Dette svarer til et nyt system med samme matrix A og ny højreside $\tilde{b}$. Løs dette system og angiv den nye løsning $\tilde{x}$.

2.4) Angiv de absolutte fejl på hvert af elementerne i den beregnede vektor. Hvad er den største absolutte fejl, og på hvilket element i vektoren forekommer denne fejl?

Spørgsmål 3. Beregningstider

Formålet med dette spørgsmål er at studere beregningstiderne for tre forskellige måder til at løse et lineært ligningssystem $A\,x = b$.

1. Brug af "backslash" `x = A\b`.
2. Brug af `inv`, dvs `x = inv(A)*b`.
3. Brug af determinanter, som beregnes med Matlab-funktionen `det`:

$$x_i = \frac{\det(A^{[i]})}{\det(A)}, \qquad i = 1, 2, \ldots, n$$

   hvor den $i$te søjle i $A^{[i]}$ er b (dette kaldes Cramer's rule i lærebøgerne).

3.1) Lav et script som, for et givet n, laver en tilfældig $n \times n$ matrix A og en tilfældig højreside b, og som bruger Matlabs `tic` og `toc` til at måle beregningstiden for de tre metoder. Dit script skal kun beregne $\det(A)$ én gang.

3.2) Brug dit script til at finde beregningstiderne for $n = 100$, 200, 400 og 800, vis resultaterne i en tabel, og kommenter resultaterne. For at determinanten af A ikke giver over- eller under-flow må matricen A ikke have for store eller for små værdier; valget $A = \frac{\mathrm{rand}(n)}{5}$ er passende her. Hvor meget hurtigere er det at bruge "backslash" i forhold til de to andre metoder, for $n = 800$?

Spørgsmål 4. Flere flops

Formålet med dette spørgsmål er at træne optælling af flops.

4.1) Et indre produkt (prikprodukt) mellem to vektorer $u, v \in \mathbb{R}^n$ er defineret ved

$$u \cdot v = \sum_{i=1}^{n} u_i v_i = u^T v, \tag{3}$$

og det beregnes i Matlab som `u'*v`. Hvor mange flops bruges til at beregne `u'*v` for vektorer af længde `n`?

4.2) Matlab-funktionen `norm` beregner 2-normen af en vektor. Hvis $x \in \mathbb{R}^n$ så er 2-normen af x defineret som

$$\|x\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}\ . \tag{4}$$

Hvor mange flops kræver det at beregne `norm(x)`?

4.3) Algoritmen nedenfor tager en `n`×`n` matrix `A` og overskriver søjlerne med vektorer som er ortonormale (det kaldes Gram-Schmidt ortonormalisering):

```
for j = 1 : n
    for k = 1 : j-1
        A(:,j) = A(:,j) - ( A(:,j)'*A(:,k) )*A(:,k);
    end
    A(:,j) = A(:,j)/norm(A(:,j));
end
```

Hvor mange flops bruger algoritmen? Du skal bruge svarene fra spørgsmål **4.1** og **4.2**, og du behøver kun at angive leddet med den højeste potens af `n`.

Spørgsmål 5. Specialiseret Gauss-elimination

Formålet med dette spørgsmål er at illustrere, hvordan beregningsarbejdet kan reduceres, når man udnytter strukturen i et beregningsproblem. En "Hessenberg-matrix" er en matrix som har nuller under den nedre bidiagonal, dvs den har formen:

$$A = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}$$

Her er Matlab-koden fra lærebogen som foretager Gauss-elimination uden pivotering på matricen `A` og højresiden `b`, samlet i array'et `[A b]`:

```
1   Aug = [A b]; n = size(A,1); nb = n+1;
2   for k = 1 : n-1
3       for i = k+1 : n
4           factor = Aug(i,k)/Aug(k,k);
5           Aug(i,k) = 0;
6           Aug(i,k+1:nb) = Aug(i,k+1:nb) - factor*Aug(k,k+1:nb);
7       end
8   end
```

5.1) Omskriv denne kode, så den udnytter strukturen af Hessenberg-matricen og dermed bliver mere effektiv for denne type af matricer. Specifikt skal linje `3-7` i din kode udnytte strukturen således at der kun regnes på ikke-0 elementer.

5.2) Hvor mange flops bruger din kode? Svaret skal gives som funktion af `n`.

Teori-spørgsmål I. LU-faktorisering

Vi betragter Gauss-elimination og LU-faktorisering *uden* pivotering. Forklar strukturen i matricerne L og U, enten i ord eller med en tegning, og forklar sammenhængen mellem disse matricer og resultatet af Gauss-eliminationen.

Teori-spørgsmål II. Bevis for LU-faktorisering

Vi betragter Gauss-elimination og LU-faktorisering *uden* pivotering. Formålet med dette spørgsmål er at illustrere at LU-faktoriseringen faktisk er en faktorisering. Dvs. vi vil vise at $L\,U = A$, for 3×3 matricer – men på en sådan måde, at beviset kan overføres til $n \times n$-matricer. Vi indfører disse fire matricer – bemærk at de tre sidste afhænger af en parameter x:

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}, \quad \mathcal{L}_{21,x} = \begin{bmatrix} 1 & 0 & 0 \\ x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathcal{L}_{31,x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x & 0 & 1 \end{bmatrix}, \quad \mathcal{L}_{32,x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & x & 1 \end{bmatrix}.$$

II.1) Vis at matricen $\mathcal{L}_{21,x}$ ved produktet $\mathcal{L}_{21,x}\,A$ udfører rækkeoperationen $R_2 := R_2 + x\,R_1$ på A.

II.2) Hvilke rækkeoperationer udfører matricerne $\mathcal{L}_{31,x}$ og $\mathcal{L}_{32,x}$ når de bliver ganget på A fra venstre?

II.3) Forklar at den del af Gauss-elimination uden pivotering som laver nuller under diagonalen på A kan udføres ved produktet

$$\mathcal{L}_{32,x_3}\,\mathcal{L}_{31,x_2}\,\mathcal{L}_{21,x_1}\,A,$$

hvis x_1, x_2 og x_3 vælges hensigtsmæssigt. Af dit svar skal det fremgå hvordan x_1, x_2 og x_3 relaterer sig til Gauss-eliminations algoritmen.

Dette betyder at U-matricen i LU-faktoriseringen kan skrives som

$$\mathcal{L}_{32,x_3}\,\mathcal{L}_{31,x_2}\,\mathcal{L}_{21,x_1}\,A = U, \tag{5}$$

og at produktet $\mathcal{L}_{32,x_3}\,\mathcal{L}_{31,x_2}\,\mathcal{L}_{21,x_1}$ må være L^{-1}, hvor L er den nedre trekants matrix som optræder i LU-faktoriseringen. For at finde L må vi altså finde de inverse til $\mathcal{L}_{32,x_3}$, $\mathcal{L}_{31,x_2}$ og $\mathcal{L}_{21,x_1}$. Det kan let vises at

$$\mathcal{L}_{21,x}^{-1} = \mathcal{L}_{21,-x}, \qquad \mathcal{L}_{31,x}^{-1} = \mathcal{L}_{31,-x}, \qquad \mathcal{L}_{32,x}^{-1} = \mathcal{L}_{32,-x}. \tag{6}$$

Dette ønskes ikke bevist (det skyldes at den inverse operation til at addere en række, er at subtrahere den igen). Matrixligningen $\mathcal{L}_{32,x_3}\,\mathcal{L}_{31,x_2}\,\mathcal{L}_{21,x_1}\,A = U$ kan nu omskrives til $A = \mathcal{L}_{21,x_1}^{-1}\,\mathcal{L}_{31,x_2}^{-1}\,\mathcal{L}_{32,x_3}^{-1}\,U$.

II.4) Vis ved en direkte udregning at

$$L = \mathcal{L}_{21,x_1}^{-1}\,\mathcal{L}_{31,x_2}^{-1}\,\mathcal{L}_{32,x_3}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -x_1 & 1 & 0 \\ -x_2 & -x_3 & 1 \end{bmatrix}, \tag{7}$$

og konkluder at hvis x_1, x_2 og x_3 vælges ud fra Gauss-eliminationen da opfylder L og U præcist beskrivelsen af LU-faktoriseringen i Teori-spørgsmål I. Så vi har vist at $A = L\,U$ og dermed forstået hvordan LU-faktoriseringen af A virker.

Teori-spørgsmål III. Cholesky-faktorisering

Hvad er forskellen på LU-faktorisering og Cholesky-faktorisering, dvs. hvornår bruger man den ene og den anden, og hvad er forskellen i regnearbejde?

Teori-spørgsmål IV. Øvre grænse for relativ fejl

Følgende vigtige formel vil altid gælde for en given matrix A og en vektor b:

$$\boxed{\|A\,b\|_2 \leq \|A\|_2 \cdot \|b\|_2 \ .} \tag{8}$$

Vi betrager to lineære ligningssystemer $Ax = b$ og $A\tilde{x} = \tilde{b}$, idet vi opfatter $\tilde{b}$ som en forstyrrelse af b. Vi antager at A er regulær hvorved A^{-1} findes. I en række trin skal vi udlede en vigtig formel for den relative fejl på løsningen. Vis først disse tre udtryk:

$$\tilde{x} - x = A^{-1}(\tilde{b} - b) \ . \tag{9}$$

$$\|\tilde{x} - x\|_2 \leq \|A^{-1}\|_2 \cdot \|\tilde{b} - b\|_2 \ . \tag{10}$$

$$\|b\|_2 \leq \|A\|_2 \cdot \|x\|_2 \ . \tag{11}$$

Brug nu disse udtryk til at vise denne vigtige formel:

$$\boxed{\frac{\|\tilde{x} - x\|_2}{\|x\|_2} \leq \mathrm{Cond}[A]\, \frac{\|\tilde{b} - b\|_2}{\|b\|_2} \ , \qquad \mathrm{Cond}[A] = \|A\|_2 \cdot \|A^{-1}\|_2 \ .} \tag{12}$$

Spørgsmål 6. Fejlvurdering

Formålet med dette spørgsmål er at træne fejlvurdering, herunder normer og konditionstal. Vi vender derfor tilbage til problemet i spørgsmål 2.

6.1) Brug Matlab-funktionen `norm` til at beregne den relative forstyrrelse af højresiden $\|\tilde{b} - b\|_2/\|b\|_2$ og den relative forstyrrelse af løsningen $\|\tilde{x} - x\|_2/\|x\|_2$.

6.2) Beregn konditionstallet Cond$[A]$ for matricen vha Matlab-funktionen `cond`, og angiv en øvre grænse for den relative fejl på løsningen, jvf. formel (12).

Spørgsmål 7. Brug af faktorisering til billedrestaurering

I dette spørgsmål skal vi arbejde med fokusering af uskarpe billeder, baseret på en matematisk model af udtværingen i billedet. Vi nøjes med at betragte kvadratiske gråtone-billeder $X \in \mathbb{R}^{n\times n}$, og vi antager at sammenhængen mellem det uskarpe billede B og de skarpe billede X kan beskrives ved:

$$B = A\,X\,A \qquad \Longleftrightarrow \qquad X = A^{-1}BA^{-1} \tag{13}$$

hvor matricen $A \in \mathbb{R}^{n\times n}$ beskriver udtværingen langs rækker og søjler i X. Du skal bruge det test-problem der ligger i mat-filen `deblur.mat`; denne fil indeholder det uskarpe billede B og udtværingsmatricen A som i denne opgave kan antages at være *symmetrisk og positivt definit.* I Matlab vises et billede `X` med:

```
imagesc(A), axis image off, colormap gray.
```

7.1) I Matlab kan vi, givet `A` og `B`, beregne `X` vha. dette simple udtryk `X = A\B/A`; men det er uhensigtsmæssigt idet Matlab beregner faktoriseringen af `A` to gange. Skriv Matlab-kode som kun bruger én faktorisering af A.

7.2) Digitale billeder indeholder ofte lidt støj, som kommer fra både lyset og kameraet. Vi kan skrive det støjfyldte og uskarpe billede som $B_{\text{noisy}} = B + E$, hvor E er en matrix med støjen; den tilsvarende løsning er $\widetilde{X} = A^{-1} B_{\text{noisy}} A^{-1}$ og det kan vises at en øvre grænse for den relative fejl på den beregnede rekonstruktion $\widetilde{X}$ kan skrives som

$$\frac{\|\widetilde{X} - X\|_f}{\|X\|_f} \le \text{Cond}[A]^2 \, \frac{\|E\|_f}{\|B\|_f}, \qquad \text{Cond}[A] = \|A\|_2 \cdot \|A^{-1}\|_2$$

hvor $\|X\|_f$ er Frobenius-normen af X – den beregnes i Matlab med `norm(X,'fro')`. Benyt denne formel til at finde den maksimale værdi af $\|E\|_f$ således at vi kan garantere at den relative fejl er mindre end 2%, dvs

$$\|\widetilde{X} - X\|_f / \|X\|_f < 0.02 \ .$$

7.3) Lav en støjmatrix `E = randn(n)` i Matlab, og skaler denne matrix således at dens Frobenius-norm $\|E\|_f$ er lig med den maksimale værdi du fandt i forrige spørgsmål. Beregn $\widetilde{X}$ og check at den relative fejl er mindre end 0.02.

Spørgsmål 8. Lineær data-fitting

Ved undersøgelse af fødevarekvaliteten af kød spiller mængden af vand en vigtig rolle. Måling af vandindholdet involverer en teknik som hedder NMR (nuclear magnetic resonance), hvormed man kan identificere forskellige typer vand i kødet. Den bagvedliggende matematiske model siger, at man skal måle et tidssignal som kan beskrives ved

$$\phi(t) = a_1 \, e^{-\lambda_1 t} + a_2 \, e^{-\lambda_2 t} + a_3 \tag{14}$$

hvor t er tiden, $\lambda_1 = 27$ og $\lambda_2 = 8$ er kendte konstanter som beskriver to forskellige typer vand, a_1 og a_2 er udtryk for mængden af disse to typer vand, og a_3 er en konstant (en såkaldt "baggrund"). I filen `NMRdata8.mat` på CampusNet ligger et datasæt med $n = 50$ sammenhørende måleresultater (t_1, y_1), (t_2, y_2), ..., (t_n, y_n), og vi skal fitte funktionen i formel (14) til disse data – dvs vi skal finde de tre ubekendte a_1, a_2 og a_3.

8.1) Vektoren a med de tre ubekendte beregnes ved at løse et overbestemt lineært ligningssystem $Z\,a \approx y$, hvor elementerne i vektoren y er værdierne for y_i fra de givne data. Giv et udtryk for elementerne i matricen Z.

8.2) Lav et Matlab-script som opstiller og løser det overbestemte lineære ligningssystem, og angiv løsningsvektoren a.

8.3) Plot det fundne fit, dvs. funktionen $\phi(t)$ med de fundne værdier af a_1, a_2 og a_3, sammen med datapunkterne. Fittet skal vises som en kontinuert og glat funktion, mens datapunkterne skal vises som selvstændige datapunkter.

8.4) Hvad er den største afvigelse mellem fit og datapunkter, dvs hvad er

$$\max |\phi(t_i) - y_i| \ , \qquad i = 1, 2, \ldots, n \ , \tag{15}$$

og for hvilken værdi af i forekommer den? Beskriv hvordan du beregner svaret.

Spørgsmål 9. Ulineær data-fitting

Vi betragter nu et andet forsøg, hvor der kun optræder én type vand samt “baggrund”, dvs. modellen er nu

$$\phi(t) = a_1\, e^{-\lambda\, t} + a_3\ . \tag{16}$$

Til gengæld véd vi ikke hvilken type vand der er til stede, og derfor er både konstanten λ samt a_1 og a_3 nu ukendte. Vi skal fitte denne model til de data som ligger i filen `NMRdata9.mat` på CampusNet og beregn a_1, λ og a_3.

9.1) Skriv en Matlab-funktion `f = fitfun(t,y,p)` som beregner summen af kvadrerede afvigelser, dvs.

$$\mathtt{f} = \sum_{i=1}^{n} \big(\phi(t_i) - y_i\big)^2\ . \tag{17}$$

Input til funktionen er to vektorer `t` og `y` af længden n med værdierne for t_i og y_i samt vektoren $\mathtt{p} = [\, a_1\, ;\, \lambda\, ;\, a_3\,]$.

9.2) Brug `fminsearch` til at beregne fittet og plot det, i lighed med spørgsmål **8.3**. Hvilken værdi af λ blev beregnet?

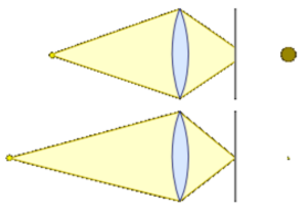
Teori-spørgsmål V. Hvad går datafitting ud på?

Forklar hvad det er vi minimerer når vi foretager datafitting, udtrykt ved datapunkterne (t_i, y_i), $i = 1, 2, \ldots, n$ samt funktionen $\phi(t)$, både i ord og med en formel (og evt. en skitse).

Teori-spørgsmål VI. Lineær og ulineær datafitting

Forklar forskellen mellem lineær og ulineær datafitting.

Opgave A. Stregkodelæseren



Vi kender stregkodelæseren (se billedet til venstre) fra mange forretninger, hvor den bruges til at aflæse stregkoder på varer (se billedet i midten). Forrest i stregkodelæseren sidder en linse, som skal fokusere lyset på en CCD der optager lyset – principielt på samme måde som i et kamera.
Hvis linsen er i fokus vil lyset fra en punktkilde ramme i ét punkt på billedplanen hvor CCD’en sidder – men hvis linsen er ude af fokus vil lyset blive spredt over et område på billedplanet, se billedet til højre. I denne opgave skal vi udvikle og implementere en algoritme som kan kompensere for en linse som er ude af fokus, og som genskaber et skarpt billede ud fra et uskarpt billede. For at gøre opgaven

simpel vil vi antage at CCD'en består af en $n \times 1$ array således at "billedet" hér kan repræsenteres som en vektor af længden n.

Vi bruger ligningen $\boxed{A\,x = b}$ som en matematisk model for sammenhængen mellem det skarpe billede $x \in \mathbb{R}^n$ (som vi ønsker at rekonstruere) og det udtværede billede $b \in \mathbb{R}^n$ som vi optager i stregkodelæseren. Elementerne i matricen $A \in \mathbb{R}^{n\times n}$ er given ved

$$a_{ij} = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(i-j)^2}{2\,\sigma^2}\right), & |i-j| \leq 10 \\ 0, & \text{ellers} \end{cases}, \qquad i,j = 1,2,\ldots,n \tag{18}$$

hvor $\sigma = 1.15$. Det fremgår af denne formel at matricen A er en *bånd-matrix*, dvs alle elementer som ligger mere end 10 elementer fra diagonalen er 0.

I denne opgave skal vi bruge en vektor `btilde` af længden $n = 500$ som ligger i filen `barcode.mat` på CampusNet, og som er et udtværet billede $\tilde{b}$ med en lille smule støj.

A.1) Lav et Matlab-script som opstiller matricen A i formel (18) og løser ligningssystemet $A\,\tilde{x} = \tilde{b}$ med højresiden fra filen `barcode.mat`. Plot både højresiden $\tilde{b}$ og den fundne løsning $\tilde{x}$.

A.2) Vi skal nu vurdere fejlen på den beregnede løsning $\tilde{x}$, som skyldes at der er små datafejl i vores højreside $\tilde{b}$, og som viser sig som små "krøller" i plottet af $\tilde{x}$. Vi oplyser at den relative fejl på data er $\|\tilde{b} - b\|_2/\|b\|_2 = 1.5 \cdot 10^{-4}$. Beregn den øvre grænse for den relative fejl på løsningen $\|\tilde{x} - x\|_2/\|x\|_2$.

Producenten af stregkodelæseren skal garantere at den relative fejl på $\tilde{x}$ er højst 1%, altså at

$$\|\tilde{x} - x\|_2/\|x\|_2 \leq 0.01\ . \tag{19}$$

Producenter skal også garantere at $\tilde{x}$ beregnes så hurtigt som muligt, hvilket kan opnås i Gauss-eliminationen ved ikke at lave beregninger på 0-elementerne.

A.3) Er formel (19) opfyldt med de givne data? Hvis ikke, hvor stor må den relative fejl på data, altså $\|\tilde{b} - b\|_2/\|b\|_2$, maksimalt være for at sikre at (19) er opfyldt?

A.4) Koden `GaussNaive.m` fra CampusNet foretager simpel Gauss-elimination uden pivotering. Du skal modificere denne kode så den udnytter strukturen i matricen, dvs. således at beregningerne kun involverer elementer i `A` der er forskellige fra 0; disse modifikationer skal foretages i både "forward elimination" og i "back substitution". NB: vi har modificeret koden en smule, i forhold til koden i lærebogen, så det er klarere hvilke operationer der foretages på hhv. `A` og `b` og således at der faktisk sættes 0'er under diagonalen i matricen `A`.

A.5) Angiv et estimat for hvor mange flops din modificerede kode bruger til at løse ligningssystemet. Svaret skal gives som funktion af n og du behøver kun angive det dominerende led (en helt præcis optælling af flops er unødigt kompliceret).

Opgave B. Kalibrering af stregkodelæseren

Producenten af stregkodelæseren bruger en linse som er billig at producere, men det betyder at udtværingen i billedet er lidt forskellig for hver stregkodelæser der produceres. Når algoritmen fra opgave **A** skal bruges i en given stregkodelæser, skal den derfor *kalibreres* således at den passer til linsen – dvs vi skal finde den værdi af parameteren σ i modellen (18) som bedst beskriver linsen.

Til at bestemme σ for en given stregkodelæser laver vi en måling med en skarp og koncentreret lyskilde, som med en ideel linse giver lys på netop én pixel. Det billede $\tilde{b}$ vi måler bliver hermed et billede af udtværingen i linsen.
Bemærk at det eksakte billede af en punktkilde i pixel nummer 10, dvs

$$x = \{\underbrace{0\ 0\ \ldots\ 0}_{9 \text{ elementer}}\ 1\underbrace{0\ 0\ 0\ 0\ \ldots\ 0}_{490 \text{ elementer}}\} \tag{20}$$

bliver udtværet svarende til højreside $b = A\,x$ givet ved den 10'ende søjle i A. Dvs.

$$b_i = a_{i,10} = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(i-10)^2}{2\,\sigma^2}\right), & |i-10| \leq 10 \\ 0, & \text{ellers} \end{cases}, \qquad i = 1, 2, \ldots, n\ . \tag{21}$$

Vi kan bruge ovenstående til at beregne σ vha ulineær data-fitting, hvor vi har 25 datapunkter $y_1, y_2, \ldots, y_{25}$ som skal svare til de første 25 elementer i vektoren b ovenfor – dog med målefejl pga diverse fejl i forsøget. Formålet er således at bestemme σ ved fitte en passende funktion til disse data.

B.1) Opskriv det ulineære data-fitting problem til bestemmelse af σ, med de 25 datapunkter og én ukendt parameter σ. Du skal udnytte at b_i i formel (21) er en funktion af σ.

B.2) Hent data (i, y_i), $i = 1, \ldots, 25$ der ligger som to vektorer `I` og `Y` i filen `kalibreringsdata.mat` på CampusNet. Brug funktionen `fminbnd` til at beregne fittet, plot fittet, og angiv den fundne værdi af σ (den svarer **ikke** til den værdi der blev brugt i opgave **A**).

Opgave C. Frobenius-normen af en matrix

Hvorledes defineres Frobenius-normen af en matrix $\|A\|_f$, og forklar hvor mange flops det "koster" at beregne denne norm for en $n \times n$ matrix?