MA2501 Numeriske metoder

Vår 2009

Prosjekt 1.

- *Utlevering:* Tirsdag 27. januar.
- *Besvarelse:* MATLAB-kode + en *kort* rapport i PDF-format. Oppgi studentnr. på alle gruppedeltakere på første side i rapporten, men *ingen navn.*

  Rapporten skal bestå av to deler:

  - Resultater fra testkjøring på et av problemene under (velg selv). Søk gjerne etter mer enn en løsning.
    NB. Resultatene skal være reproduserbare, dvs. alle relevante data må være oppgitt.
  - En beskrivelse av hvordan dere har løst problemet beskrevet under "anvendelse". Samt resultatet, selvfølgelig.
- *Innlevering:* Senest fredag 13. februar. Kode og rapport sendes som e-post til faglærer. I etterkant demonstrerer gruppene sin implementasjon av Newtons metode for faglærer.
- *Gruppestørrelse:* 1 eller 2 personer pr. gruppe.
- *Veiledning:* Øvingstimene uke 5 og 7. Dette er nok for lite, så fra 6. februar kan dere ta kontakt ved behov (1348, SBII).
- *Vurdering:* Implementasjon av Newton's metode teller 70%, anvendelsen 30%. Det blir lagt vekt på ryddighet og god dokumentasjon. For full score skal også de første to punktene under *mulige forbedringer* være implementert.

Oppgave

Oppgaven går ut på å skrive en biblioteksrutine som løser et system av ikke-lineære ligninger

$$F(x) = 0, \qquad F : \mathbb{R}^m \to \mathbb{R}^m,$$

med Newtons metode, samt å anvende denne på et ikke-trivielt problem (se anvendelse).

Spesifikasjoner for rutina:

- *Argumenter inn:*
  - Et funksjonshåndtak til en funksjon som tar inn en vektor x og returnerer vektoren $F(x)$, og om nødvendig, Jakobi-matrisa $J(x)$.
  - En vektor med startverdier x_0.
  - En feiltoleranse tol.
  - Maksimalt antall iterasjoner $Nmax$.
- *Argumenter ut:*
  - Løsningen x.
  - Antall iterasjoner nit.
  - Et flagg som sier om iterasjonen var vellykket eller ikke.
- *Dessuten:*
  - Lag en god hjelpetekst (en skal kunne bruke koden uten annen informasjon enn det blir gitt ved å skrive `help` navnet på rutina).
  - Stopp iterasjonene hvis $\max_i |f_i(x)| \leq tol$ (suksess) eller $nit \geq Nmax$ (fiasko).
  - Skriv ut en advarsel hvis iterasjonene ikke konvergerer.
  - Koden skal være selvdokumentert (med kommentarer i koden), men ikke overdriv.
- *Mulige forbedringer (ikke prioriter dette):*
  - Lag en rutine for å regne ut Jakobi-matrisen numerisk. La brukeren velge om hun vil bruke analytisk eller numerisk Jakobi-matrise.
  - La tol og $Nmax$ få standard-verdier (defaults), men la brukeren kunne overstyre disse.
  - Sjekk om iterasjonene divergerer ved å se om differansen mellom to påfølgende verdier av x øker eller avtar.
  - Legg in muligheten for en pen utskrift av løsningen i hver iterasjon.
  - Og så videre.

Noen råd og tips:

- Før du starter, vær litt stø i enkel MATLAB-programmering. Ta en kikk MATLAB-rutinene som er lagt ut, du kan få noen hint der. Du bør beherske:
  - Bruk av funksjoner, se: `function, nargin, nargout.` Husk at funksjoner i MATLAB kan ta vektorer, funksjonshåndtak, matriser, .osv. som argumenter.
  - Bruk av kontroll-setninger: `if-else, for, while, break.`
  - Vektor- og matriseregning. For eksempel at ligningssystemet $Ax = b$ løses ved `x=A\b.` Hold orden på kolonne- og radvektorer.

  Følgende funksjoner kan vise seg nyttige (se `doc` for bruk)

  - `norm, sprintf, disp.`
- Start så enkelt som mulig: Finn et ligningssystem av to ligninger, to ukjente (gjerne et du kjenner løsningen på). Gjør et par iterasjoner for hånd. Skriv MATLAB-rutinen som returnerer $F(x)$ og $J(x)$ for det gitte problemet. Test den. Skriv en løkke som utfører noen ganske få Newton-iterasjoner. Test den. Sammenlign med resultatet du fikk fra håndregninga. Får dere samme resultat? Hvis ja, gratulere, dere er i gang. Hvis nei, prøv å lokaliser akkurat hvor resultatet i MATLAB-koden avvek fra det dere regnet ut for hånd.

  En annen mulighet er å teste koden på en lineær ligning. Newton skal da konvergere i en iterasjon.

  Legg inn mer funksjonalitet etterhvert, men implementer og test en ting av gangen. Sjekk at programmet gjør det du faktisk forventer av det.
- Lær å bruke MATLAB's debugger.

Testproblemer:

P1:

$$\begin{aligned} x_1^2 - 2x_1 - x_2 + 0.5 &= 0 \\ x_1^2 + 4x_2^2 - 4 &= 0 \end{aligned}$$

P2:

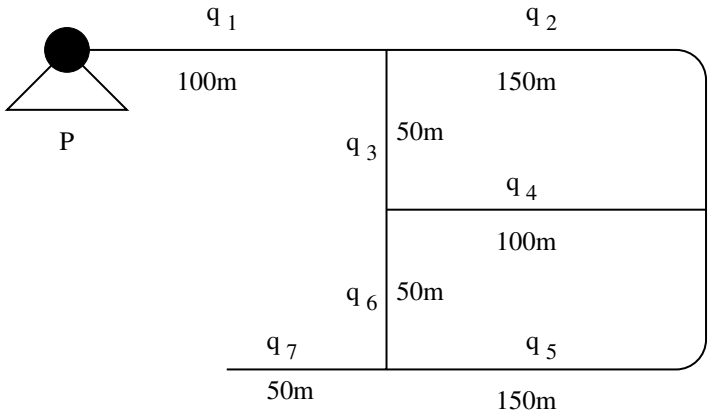
$$\begin{aligned} 5x_1^2 - x_2^2 &= 0 \\ x_2 - 0.25\left(\sin x_1 + \cos x_2\right) &= 0 \end{aligned}$$

P3:

$$\begin{aligned} 3x_1 - \cos(x_2 x_3) - 0.5 &= 0 \\ x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 &= 0 \\ e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3} &= 0 \end{aligned}$$

Anvendelse:

(Bradie -06) Vann blir pumpet gjennom gjennom et rørledningsnett, som er vist under.



Vi ønsker å finne gjennomstrømningen av vann, gitt ved q_i, $i = 1, 2 \ldots, 7$, i hver av de 7 rørledningene i nettet. Pumpen øverst til venstre pumper vann inn i nettet med et trykk på $4.1 \cdot 10^5$Pa. De fysiske lovene som bestemmer gjennomstrømningen er

- I hver av forgreiningene renner like mye vann inn som ut. I forgreininga øverst i midten gjelder altså
$$q_1 - q_2 - q_3 = 0.$$
Tilsvarende for de tre andre forgreiningene.

- I hver av rørledningene vil trykket reduseres på grunn av friksjon. Trykkfallet er gitt ved ligningen
$$\Delta P = \frac{8 f \rho L}{\pi^2 d^5} q^2$$
hvor f er en friksjonskoeffisient, ρ er vannets tetthet, L er lengden av rørledningen, q er gjennomstrømningen og d er innvendig diameter av rørledningen.

- Summen av trykkfallene rundt en sløyfe i ledningsnettet er null. Trykket er null der vannet renner ut.

Sett opp de 7 ligningene som beskriver gjennomstrømningen. La

$$f = 0.00225, \quad \rho = 998\text{kg/m}^3, \quad d = 0.15\text{m}$$

og finn q_i, $i = 1, 2 \ldots, 7$ (målt i m^3/s).