Linalg oblig 2

November 6, 2015

0.1 Linalg oblig 2

Camilla Nore
2015-10-28

0.1.1 Oppgave 1a

$$C = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix}$$

Charactheristic polynom is given by:

$$det(C - tI) = p(t)$$

$$det(C - tI) = det \begin{bmatrix} -t & 1 \\ -a_0 & -t - a_1 \end{bmatrix} = -t(-t - a_1) - (1 * -a_0) = a_0 + a_1 t + t^2$$

0.1.2 Oppgave 1b

$$C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix}$$

Charactheristic polynom is given by:

$$det(tI - C) = p(t)$$

$$det(tI - C) = det \begin{bmatrix} t & -1 & 0 \\ 0 & t & -1 \\ a_0 & a_1 & t + a_2 \end{bmatrix} = tdet \begin{bmatrix} t & -1 \\ a_1 & t + a_2 \end{bmatrix} + a_0 det \begin{bmatrix} -1 & 0 \\ t & -1 \end{bmatrix} \quad (1)$$

$$= t(t^2 + ta_2 - (-a_1)) + a_0(1 - 0) \quad (2)$$

$$= t^3 + a_2 t^2 + a_1 t + a_0 \quad (3)$$

Have shown that the characteristic polynom is p, not -p.

0.1.3 Oppgave 2i

Vi starter med å definere:

$$f(t) = f$$

Den tidsderiverte er:

$$f'(t) = f'$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} f \\ f' \\ f'' \end{bmatrix}$$

$$\mathbf{x}' = \begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} f' \\ f'' \\ f''' \end{bmatrix}$$

Vi skriver om differensiallikningen, og bruker det vi vet fra (*):

$$f''' + f'' - 4f' - 4f = 0x_3' + x_3 - 4x_2 - 4x_1 = 0x_3' = -x_3 + 4x_2 + 4x_1$$

Vi har nå et uttrykk for x_3'. Og vi ser at $x_1' = f' = x_2$. Dette kan vi skrive på matriseform:

$$\mathbf{x}' = \begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 4 & 4 & -1 \end{bmatrix} \mathbf{x}$$

Vi ser nå at dette er det samme som (**):

$$\mathbf{x}' = C\mathbf{x}$$

0.1.4 Oppgave 2 ii

Sett $f(t) = x_1(t)$, sjekk at $f(t)$ er en løsning av $(*)$
$x'(t) = Cx(t)$

$$\begin{bmatrix} x_1'(t) \\ x_2'(t) \\ x_3'(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 4 & 4 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ 4x_1 + 4x_2 - x_3 \end{bmatrix} \tag{4}$$

Dette gir oss:

$$x_1' = x_2 x_1'' = x_2' = x_3 x_1''' = x_2'' = x_3' = 4x_1 + 4x_2 - x_3 = 4x_1 + 4x_1' - x_1''$$

Setter inn for $f(t) = x_1(t)$, ser at dette løser (*).

0.1.5 Oppgave 2iii

Løsningen av et 1.ordens lineært differentiallikningsystem er gitt ved:

$$\mathbf{x}(t) = c_1 e^{\lambda_1 t}\mathbf{u}_1 + c_2 e^{\lambda_2 t}\mathbf{u}_2 + c_3 e^{\lambda_3 t}\mathbf{u}_3$$

Hvor c er konstanter, λ er egenverdiene til C, med tilhørende egenvektorer $\mathbf{u}$.

```
In [1]: import numpy as np
        C = np.asarray([[0,1,0],
                        [0,0,1],
                        [4,4,-1]])
        l, u = np.linalg.eig(C)
        print "Eigenverdier av C: ", l
        #u = u*1/u[0,0]
        #u[:,1] = u[:,1]/u[0,1]
        print "Eigenvektorer av C: \n", u
        print C
        P = u
        D = np.diag(l)
        print P.dot(C)
        print D.dot(P)
```

```
Eigenverdier av C:  [-2. -1.  2.]
Eigenvektorer av C:
[[-0.21821789 -0.57735027 -0.21821789]
 [ 0.43643578  0.57735027 -0.43643578]
 [-0.87287156 -0.57735027 -0.87287156]]
[[ 0  1  0]
 [ 0  0  1]
 [ 4  4 -1]]
[[-0.87287156 -1.09108945 -0.35913238]
 [-1.74574312 -1.30930734  1.01378605]
 [-3.49148624 -4.3643578   0.29552129]]
[[ 0.43643578  1.15470054  0.43643578]
 [-0.43643578 -0.57735027  0.43643578]
 [-1.74574312 -1.15470054 -1.74574312]]
```

```
In [2]: u1 = np.asarray([1,2,4])
        print u1 / np.linalg.norm(u1)
```

```
[ 0.21821789  0.43643578  0.87287156]
```

0.1.6 Oppgave 2 iv)

Den generelle løsniningen blir da:

$$\mathbf{x}(t) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}(t) = \begin{bmatrix} f(t) \\ f'(t) \\ f''(t) \end{bmatrix} = c_1 e^{-2t} \begin{bmatrix} -1 \\ 2 \\ -4 \end{bmatrix} + c_2 e^{-t} \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} + c_3 e^{2t} \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix} \tag{5}$$

Generell løsning finner jeg ved å bruke første rad i uttrykket over:

$$\mathbf{f}(t) = -c_1 e^{-2t} - c_2 e^{-t} + c_3 e^{2t} \tag{6}$$

Spesiell løsning:

$$\mathbf{f}(t) = -e^{-2t} + 2e^{-t} \tag{7}$$

Sjekker ved å sette inn for t = 0

$$\mathbf{f}(0) = -e^0 + 2e^0 = -1 + 2 = 1 \tag{8}$$

0.1.7 Oppgave 3i

Sjekk at v er en egenvektor for C, tilhørende λ. Det betyr at:

$$C\mathbf{x} = \lambda \mathbf{v}_\lambda \tag{9}$$
$$C\mathbf{x} - \lambda \mathbf{v}_\lambda = 0 \tag{10}$$
$$(C - I_{3x3}\lambda)\mathbf{v}_\lambda = 0 \tag{11}$$
$$\tag{12}$$

$$(C - I\lambda) = \begin{bmatrix} -\lambda & 1 & 0 & \cdots & 0 \\ 0 & -\lambda & 1 & \cdots & 0 \\ 0 & 0 & -\lambda & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} - \lambda \end{bmatrix} \begin{bmatrix} 1 \\ \lambda \\ \lambda^2 \\ \vdots \\ \lambda^{n-1} \end{bmatrix} \tag{13}$$

$$= \begin{bmatrix} -\lambda + \lambda \\ -\lambda^2 + \lambda^2 \\ -\lambda^3 + \lambda^3 \\ \vdots \\ -a_0 - a_1\lambda - a_2\lambda^2 - \cdots - a_{n-1}\lambda^{n-1} - \lambda^n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ -p(\lambda) \end{bmatrix} \tag{14}$$

Siden λ er en reel rot av p vil $p(\lambda)$ være lik null. Dermed er v en egenvektor.

0.1.8 Oppgave 3 ii

$$\mathbf{C} - \lambda_\mathbf{1}\mathbf{I} = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 4 & 4 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 0 & -1 \\ 0 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 4 & 0 & -1 \\ 0 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Setter $x_3 = -4, x_2 = 2$ og $x_1 = -1$. Dette gir

$$\mathbf{u_1} = \begin{bmatrix} -1 \\ 2 \\ -4 \end{bmatrix} \mathbf{C} - \lambda_\mathbf{2} * \mathbf{I} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 4 & 4 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Setter $x_3 = -1, x_2 = 1$ og $x_1 = -1$. Dette gir

$$\mathbf{u_2} = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} \mathbf{C} - \lambda_\mathbf{3} * \mathbf{I} = \begin{bmatrix} -2 & 1 & 0 \\ 0 & -2 & 1 \\ 4 & 4 & -3 \end{bmatrix} = \begin{bmatrix} -2 & 1 & 0 \\ 0 & -2 & 1 \\ 0 & 6 & -3 \end{bmatrix} = \begin{bmatrix} -2 & 1 & 0 \\ 0 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -4 & 0 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Setter $x_3 = 4, x_2 = 2$ og $x_1 = 1$. Dette gir

$$\mathbf{u_3} = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix} \mathbf{x}(t) = c_1 e^{\lambda_1 t}\mathbf{u_1} + c_2 e^{\lambda_2 t}\mathbf{u_2} + c_3 e^{\lambda_3 t}\mathbf{u_3} \tag{15}$$

Den generelle løsniningen blir da:

$$\mathbf{x}(t) = c_1 e^{-2t} \begin{bmatrix} -1 \\ 2 \\ -4 \end{bmatrix} + c_2 e^{-t} \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} + c_3 e^{2t} \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix} \tag{16}$$

Den spesielle løsniningen for initialbetingelsen $\mathbf{x}(0) = \begin{bmatrix} 1 \\ 0 \\ -2 \end{bmatrix}$ blir da:

$$\mathbf{x}(0) = c_1 \begin{bmatrix} -1 \\ 2 \\ -4 \end{bmatrix} + c_2 \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} + c_3 \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix} \tag{17}$$

Løser og finner $c_1, c_2 og c_3$:

$$\begin{bmatrix} -1 & -1 & 1 & 1 \\ 2 & 1 & 2 & 0 \\ -4 & -1 & 4 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 0 & 1 & 4 & -2 \\ 0 & 1 & 0 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{18}$$

Løsniningen blir da:

$$\mathbf{x}(t) = e^{-2t}\begin{bmatrix}-1\\2\\-4\end{bmatrix} + (-2)e^{-t}\begin{bmatrix}-1\\1\\-1\end{bmatrix} \tag{19}$$

0.1.9 Oppgave 3ii

Egenrommet er alle vektorer $\mathbf{x}$ som tilfredstiller $C\mathbf{x} = \lambda\mathbf{x}$. Eneste måten å oppfylle dette på er å bruke en lineær kombinasjon av $\mathbf{v}_\lambda$.

Det betyr:

$$\mathbf{x} \in E_\lambda \Leftrightarrow \mathbf{x} = \alpha\mathbf{v}_\lambda \quad \Big| \quad \alpha \in \mathbb{R}$$

Som er det samme som rommet, eller linja, som utspennes av $\mathbf{v}_\lambda$.

Med andre ord Span$\{\mathbf{v}_\lambda\}$. Rommet utspennes av en vektor, og har derfor bare en frihetsgrad, og dimensjon 1. Se eget notat.

0.1.10 Oppgave 3iii

Diagonal dekomponisjon:

$$C = P^{-1}DP$$

Hvis polynomet p ikke har n distinkte røtter, vil noen av elementene i siste rad av C, være like hverandre, og vi vil ikke lenger ha n uavhengige egenverdier.

Dermed kan vi heller ikke lage en P, med full rank, som kan inverteres.

P er sammensatt av egenvektorene til C:

$$P = \begin{bmatrix}\mathbf{v}_{\lambda_1} & \mathbf{v}_{\lambda_2} & \mathbf{v}_{\lambda_3} & \cdots & \mathbf{v}_{\lambda_n}\end{bmatrix}$$

0.1.11 Oppgave 4 - Sdrot funksjon

Programmet mitt finner en løsning for p1, men konvergerer ikke for p2.

```
In [5]: import numpy as np

        np.set_printoptions(precision=4)
        p1 = [ 1, -1, -3, -1, 2, 1]
        p2 = [1, -2, 4, -6]
        print p1[::-1]

        def sdrot(pol_coefficients, tol=.5e-6, max_iterations = 30):
            """ Find an approximation of lambda_1 for the polynomial p.

            Tar inn et reelt polynom p med ledende koeffisient 1 og
            returnerer en approksimativ verdi av λ1 når kompanion-matrisen
            C til p har en strengt dominant egenverdi λ1.
            """
            rev_coeff = pol_coefficients[::-1] # Reverse order
            rev_coeff = rev_coeff[:-1]         # Drop last element, always 1.
            p = np.asarray(rev_coeff)          # Convert to array
            n = len(p)
            C = np.zeros((n,n)) # Initialize Companion matrix of p
            C[n-1,:] = p*-1.0    # Copy p to last row
            C[:n-1,1:] = np.eye(n-1)  # Create diagonal
            x = np.random.rand(n)     # Initialize with random.
            x_values = []
            mu_values = []
```

```
    for i in range(max_iterations):
        x = C.dot(x)
        max_val = max(np.abs(x))
        max_index = np.where(np.abs(x)==max_val)[0][0]
        mu = x[max_index]
        x = 1/mu*x
        mu_values.append(mu)
        x_values.append(x)
        error = max(abs(C.dot(x)-mu*x))
        if error < tol: # Hvis feil < tol --> vi har en løsning
            return mu_values[-1]
    # For-løkka fullførte uten at vi fant en løsning
    print "Error: No solution found within max iterations."
    return mu_values[-1]

print "Dominant root of p1:\n", sdrot(p1)
print "Dominant root of p2:\n", sdrot(p2)
print "Actual roots of p1:\n", np.roots(p1)
print "Actual roots of p2:\n", np.roots(p2)
```

```
[1, 2, -1, -3, -1, 1]
Dominant root of p1:
2.29537603157
Dominant root of p2:
Error: No solution found within max iterations.
1.0
Actual roots of p1:
[ 2.2954+0.j      0.8593+0.j     -0.8172+0.5539j -0.8172-0.5539j
 -0.5202+0.j    ]
Actual roots of p2:
[ 0.1443+1.8669j  0.1443-1.8669j  1.7113+0.j    ]
```

0.1.12 Oppgave 5

```
In [6]: import scipy.linalg
        A = scipy.linalg.pascal(5)
        p3 = -np.poly(A)
        np.set_printoptions(precision=4)
        print "The Pascal matrix:\n", A
        print "The characteristic pol of A:\n", p3
        print "Dominant root of p3:", sdrot(p3)
        print "Actual roots of p3:", np.roots(p3)
        #Control by running eig(A)
        [eig, eigv] = np.linalg.eig(A)
        print "Eigenvalues of A: ", eig
```

```
The Pascal matrix:
[[ 1  1  1  1  1]
 [ 1  2  3  4  5]
 [ 1  3  6 10 15]
 [ 1  4 10 20 35]
 [ 1  5 15 35 70]]
The characteristic pol of A:
[  -1.   99. -626.  626.  -99.    1.]
Dominant root of p3: -105.017743716
```

```
Actual roots of p3: [  9.2290e+01   5.5175e+00   1.0000e+00   1.8124e-01   1.0835e-02]
Eigenvalues of A:  [  9.2290e+01   5.5175e+00   1.0000e+00   1.8124e-01   1.0835e-02]
```

0.1.13 Oppgave 6

In [7]:
```python
An =  A
np.set_printoptions(precision=4)
for i in range(8):
    [Q,R] =  np.linalg.qr(An)
    An = R.dot(Q)

print "Q:\n", Q
print "R:\n", R

# You can see that the diagonal elements of R goes to the eigen-
# values of A. The sign of the QR decomposition is implementation specific,
# and may be different between python and matlab.
```

```
Q:
[[ -1.0000e+00  -3.5644e-08   5.4476e-13  -2.0941e-18   2.1157e-27]
 [  3.5644e-08  -1.0000e+00   2.7234e-05  -1.4842e-10   2.0064e-19]
 [  4.2595e-13  -2.7234e-05  -1.0000e+00   1.0900e-05  -2.3645e-14]
 [  1.4466e-18  -1.4842e-10  -1.0900e-05  -1.0000e+00   4.9438e-09]
 [ -2.1157e-27   2.9045e-19   3.0241e-14   4.9438e-09   1.0000e+00]]
R:
[[ -9.2290e+01   3.4862e-06   4.5114e-11  -1.9560e-14  -7.7724e-16]
 [  0.0000e+00  -5.5175e+00  -1.7749e-04  -1.1426e-09   4.9722e-15]
 [  0.0000e+00   0.0000e+00  -1.0000e+00  -1.2875e-05   3.8556e-14]
 [  0.0000e+00   0.0000e+00   0.0000e+00  -1.8124e-01   9.4960e-10]
 [  0.0000e+00   0.0000e+00   0.0000e+00   0.0000e+00   1.0835e-02]]
```