Innlevering Matlab-oppgaver Øving 3 våren 2020 TDAT2002

Av Even Gultvedt

Innhold

Kapittel 2.7

CP3

Matlab-fil:

```
%Ta input [1;1] og [-1;-1] for å få de to resultatene
function x = Task27CP3(x)
    format long;

    while(1)
        s = linsolve(DF(x), -F(x));
        % Bryter hvis det vi får ut av å regne ut et steg til er mindre enn
        % ti desimaler
        if (max(abs(s)) < 5e-9)
            break;
        end
        x = x + s;
    end
end

function f = F(x)
    f1 = x(1)^3-x(2)^3+x(1);
    f2 = x(1)^2 + x(2)^2 - 1;
    f = [f1;f2];
end

function DF = DF(x)
    DF = [3*x(1)^2+1 -3*x(2)^2; 2*x(1) 2*x(2)];
end
```

Resultat:

```
>> Task27CP3([1;1])

ans =

   0.507992000407953
   0.861361786661985
```

```
>> Task27CP3([-1;-1])

ans =

  -0.507992000407953
  -0.861361786661985
```

Even Gultvedt – Matte, Øving 3 – 23/3-2020

CP5a

Matlabkode:

```
%Newtons metode til å finne:
%To punkter tre kuler har til felles
%[1/3, 1/3, 1/3] kan finnes ved å ha P=[0 0 0]
%[1 1 1] kan finnes ved å ha P=[1 1 1]
function P = Task27CP5a(P)
    C = [1 1 0; 1 0 1; 0 1 1;];
    R = [1;1;1];

    while(1)
       A = Dr(C, P);
       s = -(A'*A)\(A'*r(C, R, P));
       P = P + s';
       if (max(abs(s)) < .5e-6)
           break;
       end
    end

end

function S = S(C, P)
    S = sqrt(sum((C - P).^2, 2));
end

function r = r(C, R, P)
    r = S(C, P) - R;
end

function Dr = Dr(C, P)
    Dr = (P-C)./S(C, P);
end
```

Resultat:

```
>> Task27CP5a([0 0 0])

ans =

   0.333333333333333   0.333333333333333   0.333333333333333
```

```
>> Task27CP5a([1 1 1])

ans =

     1     1     1
```

CP5b

Matlabkode:

```
%Newtons metode til å finne:
%To punkter tre kuler har til felles
%[ 1 2 3] kan finnes ved å ha P=[0 0 0]
%[1.888 2.444 2.111] kan finnes ved å ha P=[1 1 1]
function P = Task27CP5b(P)
    C = [1 -2 0; -2 2 -1; 4 -2 3;];
    R = [5;5;5];

    while(1)
       A = Dr(C, P);
       s = -(A'*A)\(A'*r(C, R, P));
       P = P + s';
       if (max(abs(s)) < .5e-6)
           break;
       end
    end

end

function S = S(C, P)
    S = sqrt(sum((C - P).^2, 2));
end

function r = r(C, R, P)
    r = S(C, P) - R;
end

function Dr = Dr(C, P)
    Dr = (P-C)./S(C, P);
end
```

Resultat:

```
>> Task27CP5b([0 0 0])

ans =

   0.999999999999885   1.999999999999942   3.000000000000114
```

```
>> Task27CP5b([1 1 1])

ans =

   1.888888888888887   2.444444444444443   2.111111111111114
```

Kapittel 4.1 – CP3

Matlabkode: (Anbefaler å zoome inn på skjermen, bildet er tydelig nok til det).

```matlab
function [] = Task41CP3()
    P = [1960 3039585530;
        1970 3707475887;
        1990 5281653820;
        2000 6079603571];
    %Fra wikipedia,
    %brukes bare for å sjekke om utregnigner er ca riktig
    actual1980 = 4434682000.00;

    [cLine, RMSELine] = fitLine(P);
    fprintf('Best fitting line: y=%.3f + %.3fx\n', cLine(1), cLine(2));
    fprintf('RMSE of line: %.3f\n', RMSELine);
    line1980 = cLine(1) + cLine(2)*1980;
    fprintf('y_line(1980)=%.3f | Error: %.3f | Relative forward error:%.3f\n\n', line1980, abs(line1980 - actual1980), abs(line1980 - actual1980) / actual1980);

    [cParabola, RMSEParabola] = fitParabola(P);
    fprintf('Best fitting line: y=%.3f + %.3fx + %.3fx^2\n', cParabola(1), cParabola(2), cParabola(3));
    fprintf('RMSE of line: %.3f\n', RMSEParabola);
    parabola1980 = cParabola(1) + cParabola(2)*1980 + cParabola(3)*1980^2;
    fprintf('y_parabola(1980)=%.3f | Error: %.3f | Relative forward error:%.3f\n\n', parabola1980, abs(parabola1980 - actual1980), abs(parabola1980 - actual1980) / actual1980);

    fprintf('Results: The parabola is the best fit, with a low relative forward error, and almost half the RMSE.');
end

function [c, RMSELine] = fitLine(P)
    A = [1 P(1, 1); 1 P(2, 1); 1 P(3, 1); 1 P(4, 1)];
    b = [P(1, 2); P(2, 2); P(3, 2); P(4, 2)];
    c = linsolve(A'*A, A'*b);
    RMSELine = RMSE(A, b, c, 4);
end

function [c, RMSEParabola] = fitParabola(P)
    A = [1 P(1, 1) P(1, 1)^2; 1 P(2, 1) P(2, 1)^2; 1 P(3, 1) P(3, 1)^2; 1 P(4, 1) P(4, 1)^2];
    b = [P(1, 2); P(2, 2); P(3, 2); P(4, 2)];
    [c, RMSEParabola] = fit(A, b);
end

function [c, RMSE_value] = fit(A, b)
    c = linsolve(A'*A, A'*b);
    RMSE_value = RMSE(A, b, c, 4);
end

function RMSE = RMSE(A, b, c, n)
    r = b - A*c;
    SE = norm(r)^2;
    RMSE = sqrt(SE/n);
end
```

Resultater:

```
>> Task41CP3()
Best fitting line: y=-147026357795.052 + 76542140.150x
RMSE of line: 36751088.162
y_line(1980)=4527079702.000 | Error: 92397702.000 | Relative forward error:0.021

Best fitting line: y=702727591849.498 + -781849921.529x + 216765.672x^2
RMSE of line: 17129714.187
y_parabola(1980)=4472888283.965 | Error: 38206283.965 | Relative forward error:0.009

Results: The parabola is the best fit, with a low relative forward error, and almost half the RMSE.
```

Kapittel 4.2 – CP3

Matlabkode:

```
function [] = Task42CP3()
    P = [1960 3039585530;
        1970 3707475887;
        1990 5281653820;
        2000 6079603571];

    actual1980 = 4434682000.00;
    [cExp, RMSEExp] = fitExp(P);

    fprintf('Best fitting exponential function: y=%.9fe^(%.3fx) \n', cExp(1), cExp(2));
    fprintf('RMSE of function: %.3f\n', RMSEExp);
    line1980 = cExp(1)*exp(cExp(2)*1980);
    fprintf('y_exp(1980)=%.3f | Error: %.3f | Relative forward error:%.3f\n\n', line1980, abs(line1980 - actual1980), abs(line1980 - actual1980) / actual1980);
end

function [cExp, RMSEExp] = fitExp(P)
    A = [1 P(1, 1); 1 P(2, 1); 1 P(3, 1); 1 P(4, 1)];
    b = [log(P(1, 2)); log(P(2,2)); log(P(3,2)); log(P(4,2))];
    [cExp, RMSEExp] = fit(A, b);
    cExp(1) = exp(cExp(1));
    disp(cExp);
    disp(RMSEExp);
end

function [c, RMSE_value] = fit(A, b)
    c = linsolve(A'*A, A'*b);
    RMSE_value = RMSE(A, b, c, 4);
end

function RMSE = RMSE(A, b, c, n)
    r = b - A*c;
    SE = norm(r)^2;
    RMSE = sqrt(SE/n);
end
```

Resultater:

```
>> Task42CP3()
Best fitting exponential function: y=0.000004726e^(0.017x)
y_exp(1980)=4361485915.410 | Error: 73196084.590 | Relative forward error:0.017
```

Kapittel 4.5 – CP1a

Matlabkode:

```
%Beste midtpunkt mellom de tre sirklene er (0.410623, 0.055501)
function x = Task45CP1()
    x = [0;0];
    while(1)
        A = Dr(x);
        r = [r1(x); r2(x); r3(x)];
        s = linsolve(A'*A, -A'*r);
        if (abs(max(s)) < 5e-10)
            break;
        end
        x = x + s;
    end
end

function r1 = r1(x)
    r1 = sqrt(x(1)^2+(x(2)-1)^2)-1;
end

function r2 = r2(x)
    r2 = sqrt((x(1)-1)^2 + (x(2)-1)^2) -1;
end

function r3 = r3(x)
    r3 = sqrt(x(1)^2+(x(2)+1)^2) - 1;
end

function Dr = Dr(x)
    Dr1 = [(x(1) / sqrt(x(1)^2+(x(2)-1)^2)), ((x(2)-1) / sqrt(x(1)^2+(x(2)-1)^2))];
    Dr2 = [(x(1)-1) / sqrt((x(1)-1)^2 + (x(2)-1)^2), (x(2)-1) / sqrt((x(1)-1)^2+(x(2)-1)^2)];
    Dr3 = [x(1) / sqrt(x(1)^2+(x(2)+1)^2), (x(2)+1) / sqrt(x(1)^2 + (x(2)+1)^2)];
    Dr = [Dr1; Dr2; Dr3];
end
```

Resultater: (Vektoren kommer som [x; y])

```
>> Task45CP1()

ans =

   0.410623197830426
   0.055501397272634
```