Norges teknisk–naturvitenskapelige universitet
Institutt for datateknikk og informasjonsvitenskap (IDI)

TDT4105 IT Grunnkurs
Høst 2008

Løsningsforslag øving 6

Denne øvingen er delvis koordinert med øving 6 i tma4100. Oppgave 1 består av å lage en funksjon for å finne normen til en partisjon (H-W-T betyr *Hass-Weir-Thomas, University Calculus*).

I oppgave 2 skal vi programmere Koch's snøfnugg. Det har ikke direkte tilknytning til tma4100/6. Du kan lese mer om om Koch's snøfnugg ved å søke på Koch snowflake.

1 Vi skal lage en funksjon for å finne partisjonsnormen $||P||$ (det lengste intervallet) til en inndeling $x \in [a, b]$ i intervaller $a = x_0 < x_1 < x_2 \ldots < x_{n-1} < x_n = b$.

a) Kall funksjonen `partition_norm`. Funksjonen skal ta inn ei liste av punkter, `points`, og levere fra seg partisjonsnormen $||P||$ (ett tall). Det er *ikke* nødvendig å sjekke om punktene i lista ligger i stigende rekkefølge. Du skal ikke bruke innebygde funksjoner som `max`, `sum`, og tilsvarende.

Løsning:

```
function norm = partition_norm(points)
    n = length(points);
    norm = 0;

    for i=2:n
        dx = points(i) - points(i-1);
        if (dx > norm)
            norm = dx;
        end
    end
end
```

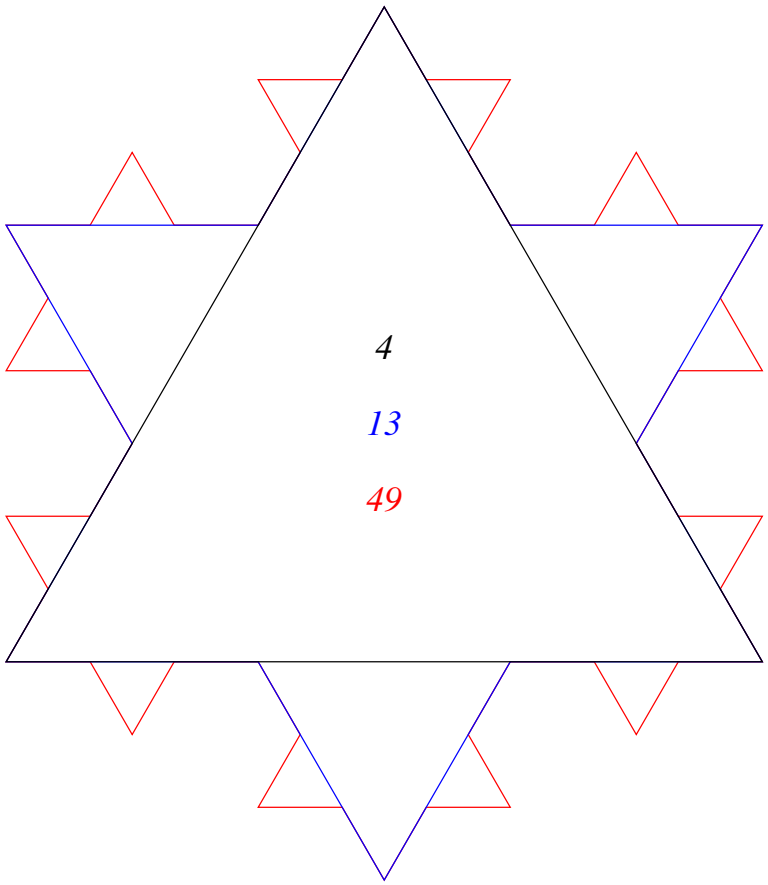
b) Sjekk funksjonen på H-W-T 5.2.34: $P = \{-2, -1.6, -0.5, 0, 0.8, 1\}$.

Løsning:

```
>> partition_norm([-2 -1.6 -0.5 0 0.8 1])
ans = 1.1000
```

2 I denne oppgaven skal vi skrive en funksjon `snowflake` for å regne ut x- og y-koordinater til punkter i Koch's snøfnugg. Koch's snøfnugg starter med 3 linjestykker (4 punkter). Deretter deles hvert linjestykke i 3, og en likesidet trekant med kanter lik $1/3$ av lengden settes inn midt på. Figur 1 viser de 3 første inndelingene. Den sorte trekanten er det første linjestykket med 4

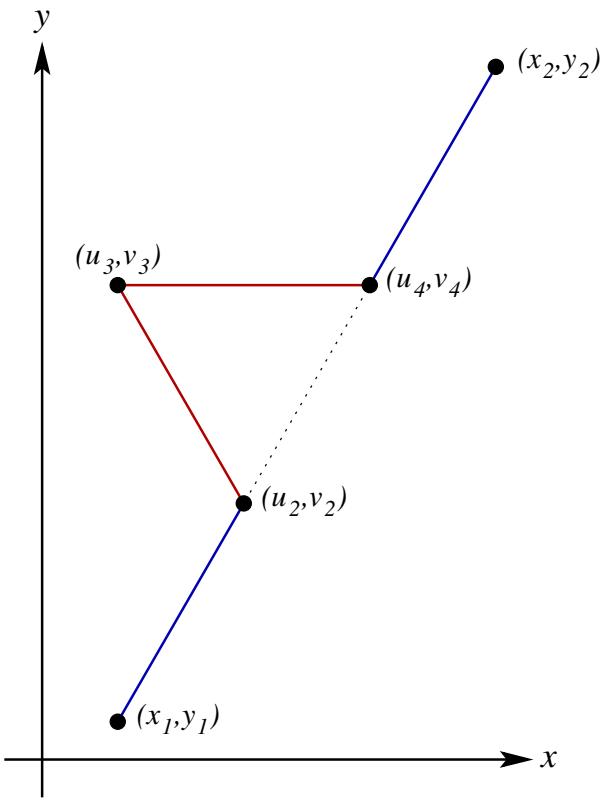
punkter. Blå linje er neste nivå med 13 punkter, så følger rødt med 49 punkter. Antallet punkter på nivå $i+1$ er $N_{i+1} = 4(N_i - 1) + 1$.



Figur 1: De 3 første nivåene i Koch's snøfnugg.

Formel (1)-(6) viser hvordan de nye punktene på ei linje regnes ut. Anta at vi setter inn en trekant på linjestykket mellom (x_1, y_1) og (x_2, y_2) som vist i figur 2. Dersom vi definerer avstandene $\Delta x = (x_2 - x_1)/3$ og $\Delta y = (y_2 - y_1)/3$ kan vi finne koordinatene til de nye punktene fra

$$u_2 = x_1 + \Delta x \tag{1}$$
$$u_3 = x_1 + \frac{3}{2}\Delta x - \frac{1}{2}\sqrt{3}\Delta y \tag{2}$$
$$u_4 = x_1 + 2\Delta x \tag{3}$$
$$v_2 = y_1 + \Delta y \tag{4}$$
$$v_3 = y_1 + \frac{3}{2}\Delta y + \frac{1}{2}\sqrt{3}\Delta x \tag{5}$$
$$v_4 = y_1 + 2\Delta y \tag{6}$$



Figur 2: Innsetting av en ny trekant i et linjestykke (x_1, y_1) - (x_2, y_2).

Når man skal teste ut en ny algoritme lages ofte et skript som fyller inn noen fornuftige verdier, kaller funksjonen man har skrevet og skriver ut noen resultater etter at den har kjørt så man kan se om ting fungerer som det skal. Dette kalles et *driverprogram* i datateknikken.

I denne øvingen får du vedlagt et slikt driverprogram, `snow_main.m`. Du skal bruke dette skriptet for å teste ut og vise fram programmeringen du har gjort.

a) Skriv funksjonen `snowflake()`. Den skal ta inn to lister `x,y` med `N` punktverdier i xy-planet som definerer en lukket kurve (polygon), dvs. at $(x_1, y_1) = (x_N, y_N)$. Funksjonen skal returnere to nye lister `u,v` av lengde $4(N-1)+1$ hvor det er satt inn trekanter i alle linjestykkene $((x_i, y_i), (x_{i+1}, y_{i+1}))$, som beregnet etter formlene (1)-(6). Bruk `snow_main` for å sjekke at `snowflake()` fungerer korrekt.

Løsning:

```
%
% Assume closed input vectors, with x(n) == x(1)
%
function [u,v] = snowflake(x,y)
    n = length(x)-1;
    u = zeros(1,4*n+1);
    v = zeros(1,4*n)+1;

    j = 1;
    for i = 1:n
x1 = x(i);
x2 = x(i+1);
y1 = y(i);
y2 = y(i+1);
rx = (x2-x1)/3;
ry = (y2-y1)/3;

u(j) = x1;
u(j+1) = x1 + rx;
u(j+2) = x1 + 1.5*rx - 0.5*sqrt(3)*ry;
u(j+3) = x1 + 2*rx;

v(j) = y1;
v(j+1) = y1 + ry;
v(j+2) = y1 + 1.5*ry + 0.5*sqrt(3)*rx;
v(j+3) = y1 + 2*ry;

j = j + 4;
    end

    % assumes x(1) == x(n+1), y(1) == y(n+1)
    u(j) = x(n+1);
    v(j) = y(n+1);
end
```

b) Koch's snøfnugg har et endelig areal, men en uendelig omkrets ($O \rightarrow \infty$). Hvordan vil dette begrense hvor detaljert du kan beregne snøfnugget ?

Løsning:

Uansett hvor mye minne du har på maskina di vil den løpe tom for minne hvis du prøver å beregne snøfnugget detaljert nok. Dersom du ønsker å zoome inn uten noen grense må du finne en annen måte å beregne utsnittet på.

Driver

```
u0 = 1; v0 = 3.1; % lower left position of plot

k = -1;
while (true)
    k = k + 1;
    if (k == 0)
        u = u0 + [0 5 10 0];
        v = v0 + [0 5*sqrt(3) 0 0];
    else
        disp(['Computing # ' num2str(k)]);
        [u,v] = snowflake(u,v);
    end

    plot(u,v); grid on; axis([0 12 0 12], 'square');
    disp(['Done with ' num2str(k) ...
          ', hit any key to continue (^C breaks) ...']);
    pause
end
```