

Bonusuppgift LANA, Block 1 Uppg. 4

Simon Sigurdhsson

20 april 2009

1 MATLAB-funktionen compress

```
function out=compress(v, epsilon)
A = [1/2 1/2]; B = [1/2 -1/2]; Z = [0 0];
[no s] = size(v);
A_1 = [wavelet_ext(eye(s/2), A); wavelet_ext(eye(s/2), B)];
A_2 = [wavelet_ext(eye(s/4), A) zeros(s/4, s/2);
       wavelet_ext(eye(s/4), B) zeros(s/4, s/2);
       zeros(s/2, s/2) eye(s/2)];
A_3 = [wavelet_ext(eye(s/8), A) zeros(s/8, 3*s/4);
       wavelet_ext(eye(s/8), B) zeros(s/8, 3*s/4);
       zeros(3*s/4, s/4) eye(3*s/4)];
out=A_3*(A_2*(A_1*v'));
for i=1:size(out)
    if abs(out(i))<epsilon
        out(i)=0;
    end
end
out = out';
```

Det finns även en funktion `inflate` som ser i princip likadan ut, med skillnaden att den utför processen baklänges (dvs. ”dekomprimerar”) med hjälp av gausselimination (MATLABs backslash). Den innehåller inte den sista biten som tar bort små värden. Relevant är även funktionen `wavelet_ext` som är en hjälpfunktion för att dynamiskt kunna skapa de tre matriser som används i wavelet-approximationen:

```
function out=wavelet_ext(in, with)
tout = [zeros(size(in)); zeros(size(in))];
for k=1:size(in).^2
```

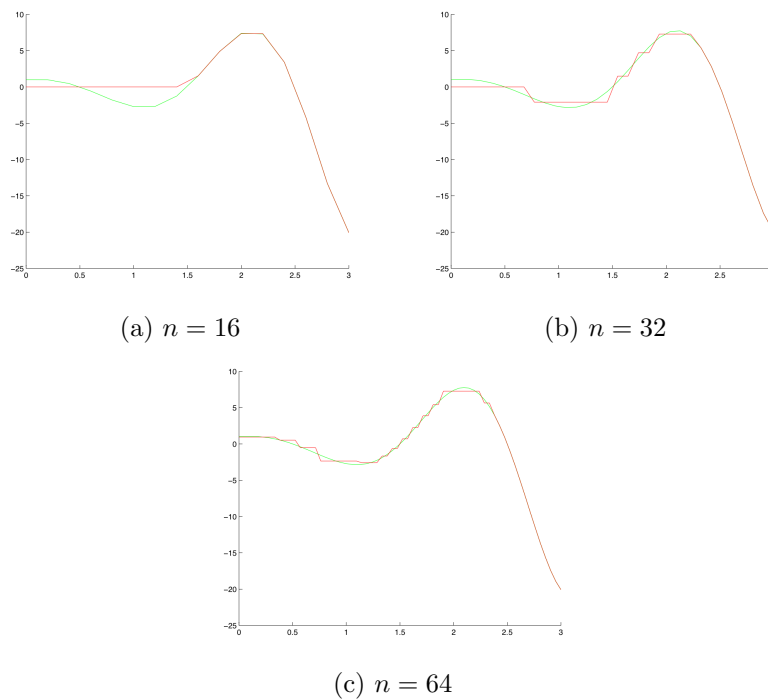
```

        i = (k*2-1);
        tout(i:i+1) = with*in(k);
    end
    out = tout';

```

2 Figurer över approximationerna

Nedan finns figurer för approximationerna. Rött är approximationen och grönt är de funktionsvärden som används.



Figur 1: Waveletapproximation av $e^x \cos(\pi x)$ på intervallet $[0, 3]$ med olika antal värden n .

3 Fel och komprimeringsgrad

	$n = 16$	$n = 32$	$n = 64$
Fel	0.2927	0.1094	0.0407
Komprimeringsgrad	0.4375	0.3750	0.4062