

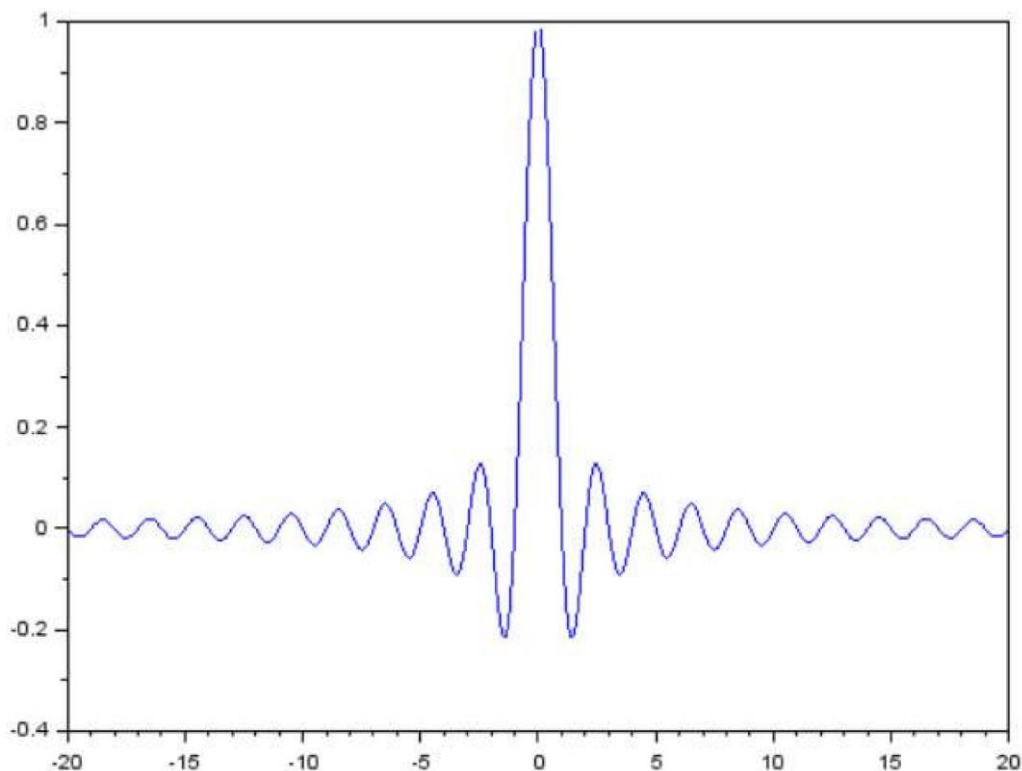
Opgaven

Voor iedere opgave geef je de code en grafische weergave weer in een pdf-bestand. Je stuurt ook voor iedere opgave het codebestand mee. Code kan je schrijven in SciNotes en bewaren. Een scilab codebestand heeft de extensie “.sce”.

1. Teken in scilab de functie $y = \sin(\pi x) / \pi x$ in voor x van -20 tot 20. Dit is een zogenaamde sinc-functie. Varianten op deze functie vind je terug bij signaalverwerkingsmethode. Bij het maken van deze opdracht hou je best rekening met het delen van 0. Als dit voorkomt vervang je de uitkomst met de waarde 1.

Oplossing:

```
x = -20:0.1:20;
if x == 0 then
    y = sin((%pi*x))/(%pi*1);
else
    y = sin((%pi*x))./(%pi*x);
end
plot(x,y)
```



2. Ontwerp een functie die een vergelijking van de tweede graad kan inlezen, de nulpunten bepalen en deze functie kan tekenen? Vang hierbij het eventueel delen door 0 op door een melding te geven “opgelet deling door 0” Zorg ervoor dat je code in alle omstandigheden werkt (tip hiervoor maak je best een analyse)

Een vergelijking van de tweede graad heeft volgende vorm: $y = ax^2 + bx + c$. Met a, b en c reële getallen.

Oplossing:

```
clf;
function vergelijking(a, b, c);
    x=-50:50

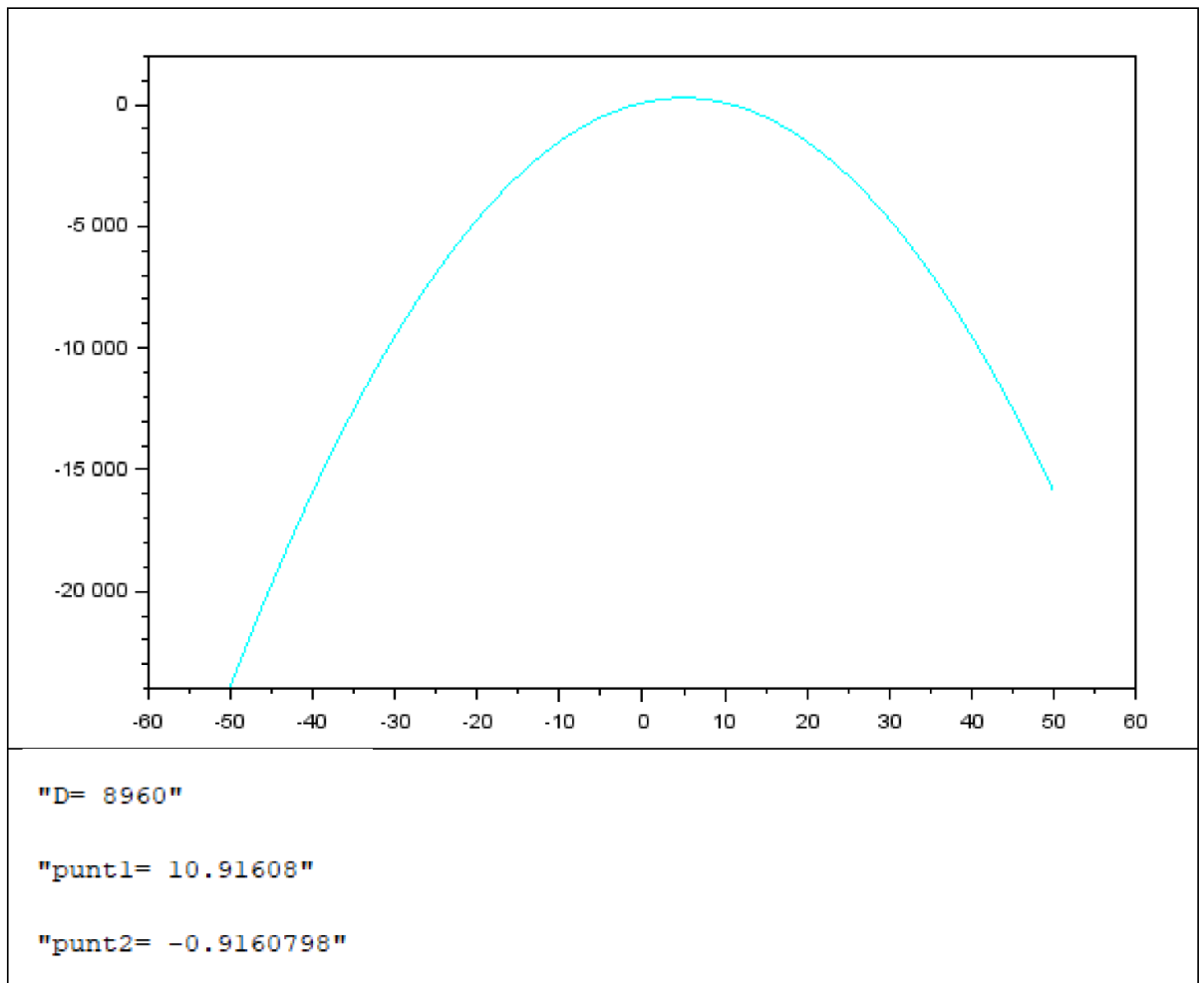
    if x == 0 then
        y=1
    else
        y = a*(x^2)+b*x+c
    end

    D=(b^2)-4.*a.*c;

    disp("D= " + string(D))
    if a == 0 then
        disp("delen door 0 gaat niet. Eerste graad getekend")
    else
        if D > 0 then
            punt1=(-b-sqrt(D))./(2*a)
            punt2=(-b+sqrt(D))./(2*a)
            disp("punt1= " + string(punt1))
            disp("punt2= " + string(punt2))
        elseif D==0 then
            punt=(b^2)./(2.*a)
            disp("punt= " + punt)
        else
            disp("geen snijpunten")
        end
    end

    plot(x,y,"c");
endfunction

vergelijking(-8,80,80)
```



3. In een digitaal systeem wordt een sinus met amplitude 2 V ingelezen met een 10-bit ADC en een samplefrequentie van 8 kHz. De sinus is gesupponeerd op een DC-spanning van 2,5 V. Stel dat er vier perioden van het signaal worden ingelezen.
- Gevraagd:
- Geef de ingevoerde digitalisatie grafisch weer in een grafiek (samplewaarde in functie van de samplenummer) (bv code 255 op moment dat vierde sample wordt ingelezen). Pas de benaming van de assen van de grafiek aan als volgt: titel grafiek : visualisatie samples; Y-as : kwantisatiewaarde sample; X-as : samplevolgnummer
 - Zet de grafiek van a) om in een amplitude in functie van de tijd grafiek. Pas de benaming van de assen van de grafiek aan als volgt: titel grafiek : visualisatie gekwantificeert ingangssignaal; Y-as : spanningswaarde
 - Bepaal de gemiddelde waarde van het digitaal ingelezen signaal
 - Bepaal de amplitude variatie van het ingelezen signaal

Oplossing:

```
clf
```

```
offset=2.5
```

```
amplitude=2
```

```
sampletekst="sample nummer"
```

```
samplewaarde="sample waarde"
```

```
//het binnenkomende signaal
```

```
t= 0.0:0.000125:0.001*4//000125 -> T van 8kHz 0.001 -> T van 1kHz
```

```
sig1kHz=(sin(2*%pi*1000*t)*amplitude+offset) //1kHz genereren
```

```
sigkwanti=sig1kHz*204.6000014
```

```
//a)
```

```
subplot(221)
```

```
plot((t*(8000)),sigkwanti)
```

```
title("visualisatie samples")
```

```
ylabel("kwantisatiewaarde sample")
```

```
xlabel("samplevolgnummer")
```

```
// 5/1023=0.004887 -> 1 = 0.004887 * x -> x = 204.6000014
```

```
// als we dan op 0.5V zitten -> digitale code is 0.5/0.004887 = 102. Dit komt
```

```
// overeen met de grafiek.
```

```
//b)
```

```
subplot(223)
```

```
plot(t,sig1kHz)
```

```
title("visualisatie gekwantificeert ingangssignaal")
```

```
ylabel("spanningswaarde")
```

```
xlabel("tijd (s)")
```

```
//c)
```

```
gem=mean(sigkwanti)
```

```
disp("gemiddelde waarde digitaal: " + string(gem))
```

```
//d)
```

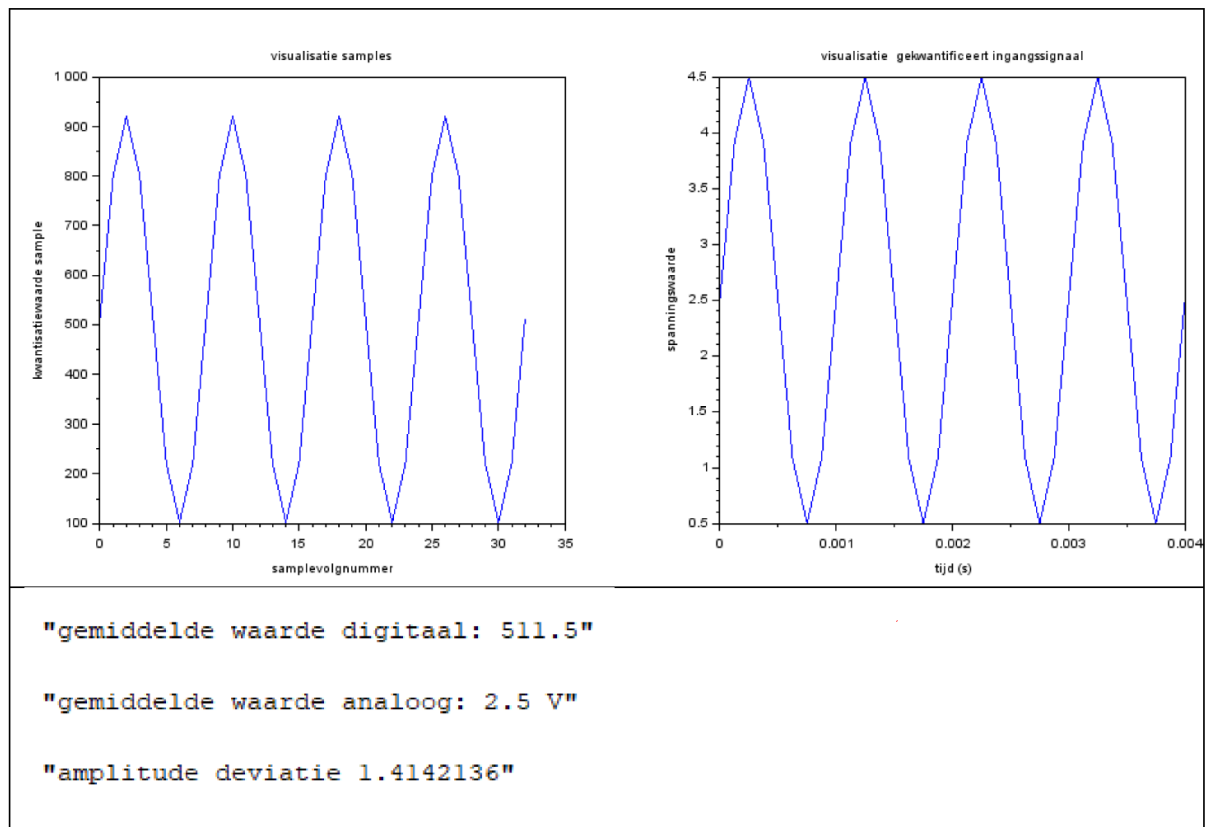
```
gem2=mean(sig1kHz)
```

```
disp("gemiddelde waarde analoog: " + string(gem2) + " V")
```

```
//e)
```

```
deviatie=stdev(sig1kHz,'c')
```

```
disp("deviatie " + string(deviatie))
```



4. Maak een grafiek waarin een uniform randomsignaal (in het groen) met 200 randomwaarden gelegen tussen de waarde 0 en 4. Iedere randomwaarde stelt een bepaalde amplitude voor. Gevraagd:
- Bepaal het maximum en minimum van dit signaal en geef weer in de grafiek waar deze zich bevinden aan de hand van een asterisk (*) in het rood.
 - Bepaal de "DC-component" van dit signaal en maak deze zichtbaar in een grafiek via een blauwe lijn. De DC-component kan gevonden worden door het gemiddelde te bepalen van dit signaal (kijk of hiervoor geen instructie is)
 - Bepaal de deviatie van het signaal tot deze "DC-component". Welk is de gemiddelde amplitudegrootte van dit signaal. Kijk of je voor deviatie geen instructie vindt. Deviatie geeft de afwijking weer van het randomgetal ten opzichte van het gemiddelde. Op deze wijze geeft de deviatie de "gemiddelde" amplitude weer van het random signaal

Oplossing:

clf

```
y=grand(1,200,"unf",0,4)
x=0:199
plot(x,y,"g");
```

```
//a)
maximum=max(y)
minimum=min(y)
maxX=find(y==maximum)
minX=find(y==minimum)
plot(maxX,maximum,"*r")
plot(minX,minimum,"*r")
disp("minimum " + string(minimum))
disp("maximum " + string(maximum))
```

```
//b)
gem=mean(y)
plot([0,200],[gem,gem],"b")
disp("gemiddelde " + string(gem))
```

```
//c)
deviatie=stdev(y,'c')
disp("deviatie " + string(deviatie))
```

