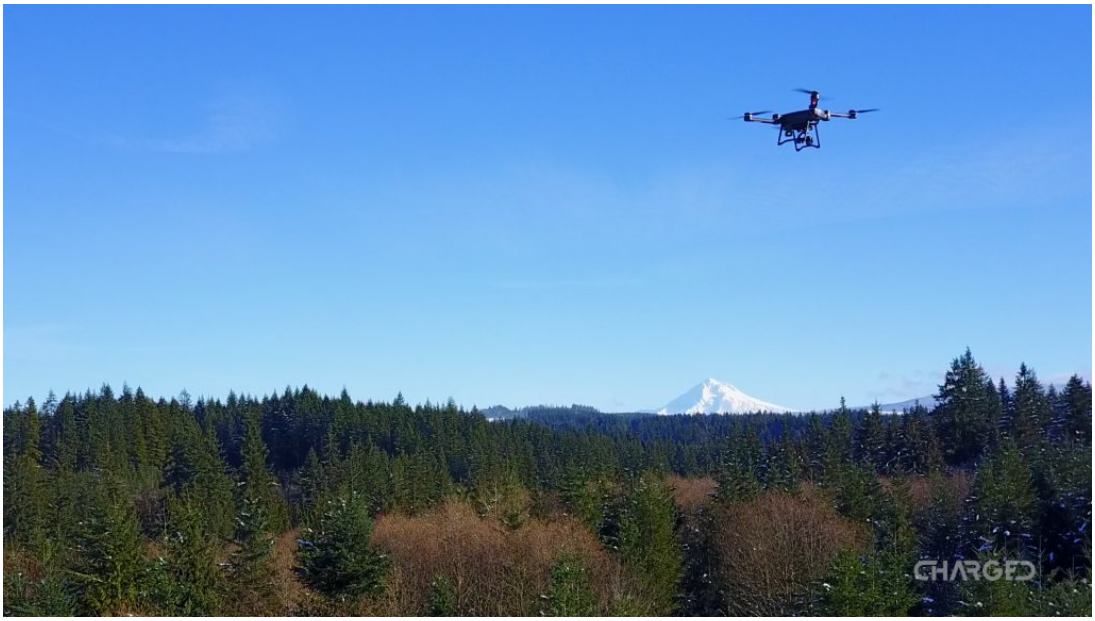
SCENARIO:
- vi skriver kod som skall styra en automatiserad drönaren
- dronen som övervakar ett bägrensat skogs område där det finns starka vindar
- vi vill att drönaren flyger på en och samma höjd hela tiden; höjden behöver justeras hela tiden
- funktionen f kontrollerar motorns styrka
- input till funktionen, x, är höjden från jorden (i enheter av den naturliga avstån-det vi vill ha drönaren i, kanske 50m)
- output, värde av f, kontrollerar motorns vavtal (i enheter som är naturliga för inbygda motorer, kanske 15000 rpm).




```
f[x_] :=
 Module[
  {a, b, c},
  a = -x^2 + 3 x - 2;
  b = x - 1;
  c = a / b;
  c
 ]
```

dronen ligger ovanför den ideela höjden

```
f[1.1]
```

```
0.9
```

dronen ligger änu högre

```
f[1.5]
```

```
0.5
```

dronen ligger under den ideela höjden

```
f[0.9]
```

```
1.1
```

dronen ligger änu mera under

```
f[0.5]
```

```
1.5
```

dronen ligger precis på höjden den skall vara

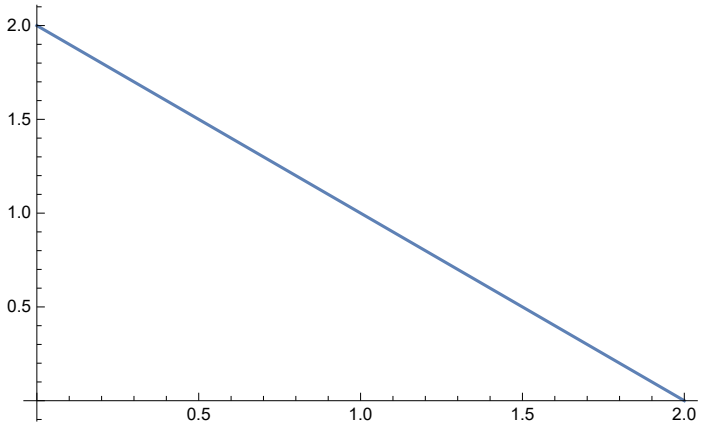
```
f[1]
```

```
Power::infy : Infinite expression 1/0 encountered. >>
```

```
Infinity::indet : Indeterminate expression 0 ComplexInfinity encountered. >>
```

```
Indeterminate
```

```
Plot[f[x], {x, 0, 2}]
```




```
fnew[x_] :=
 Module[
  {a, b, c, fnewi1 = "vadSkallViTa?"},
  a = -x^2 + 3 x - 2;
  b = x - 1;
  If[b == 0, Print["speciell fall"]; Return[fnewi1]];
  c = a / b;
  c
 ]
```

vi kan använda identitet: f(x) = -x^2+3x-4=(2-x)(x-1) och då vet vi att (-x^2+3x-4)/(x-1)=(2-x)(x-1)/(x-1)=2-x så om x är inte 1 då f(x)=2-x. vi också ser att om x närmar sig 1 då f närnar sig 2-1=1.

```
fnew[x_] :=
 Module[
  {a, b, c, fnewi1 = 1},
  a = -x^2 + 3 x - 2;
  b = x - 1;
  If[b == 0, Print["speciell fall"]; Return[fnewi1]];
  c = a / b;
  c
 ]
```

```
fnew[1.1]
```

```
0.9
```

```
fnew[1]
```

```
speciell fall
1
```

man kan undra varför allt detta krångel. vi vet ju, efter lite algebra, att f[x]=2-x, så varför inte skriva en sådan kod? sant, men det finns funktiner där man kan inte använda algebra för att förankla funktionen.

```
g[x_] :=
 Module[
  {a, b, c},
  a = Sin[x];
  b = x;
  c = a / b;
  c // N
 ]
```

```
xs = {1, 0.9, 0.5, 0.1, 0.01, 0.005, 0.0001}
```

```
{1, 0.9, 0.5, 0.1, 0.01, 0.005, 0.0001}
```

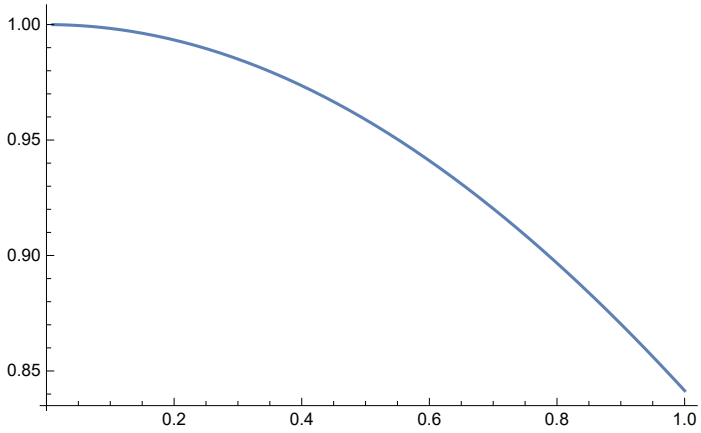
```
gs = Map[g, xs]
```

```
{0.841471, 0.870363, 0.958851, 0.998334, 0.999983, 0.999996, 1.}
```

```
{xs, gs} // Transpose // MatrixForm
```

```
⎛   1    0.841471⎞
⎜  0.9   0.870363⎟
⎜  0.5   0.958851⎟
⎜  0.1   0.998334⎟
⎜ 0.01   0.999983⎟
⎜0.005   0.999996⎟
⎝0.0001     1.   ⎠
```

```
Plot[g[x], {x, 0.01, 1}]
```




```
gnew[x_] :=
 Module[
  {a, b, c, gnewi0 = "VadSkallViTa?"},
  a = Sin[x];
  b = x;
  If[b == 0, Print["speciell fall"]; Return[gnewi0]];
  c = a / b;
  c // N
 ]
```

```
gnew[0.1]
```

```
0.998334
```

```
gnew[0]
```

```
speciell fall
VadSkallViTa?
```

```
gnew[x_] :=
 Module[
  {a, b, c, gnewi0 = 1},
  a = Sin[x];
  b = x;
  If[b == 0, Print["speciell fall"]; Return[gnewi0]];
  c = a / b;
  c // N
 ]
```

```
gnew[0]
```

```
speciell fall
1
```