

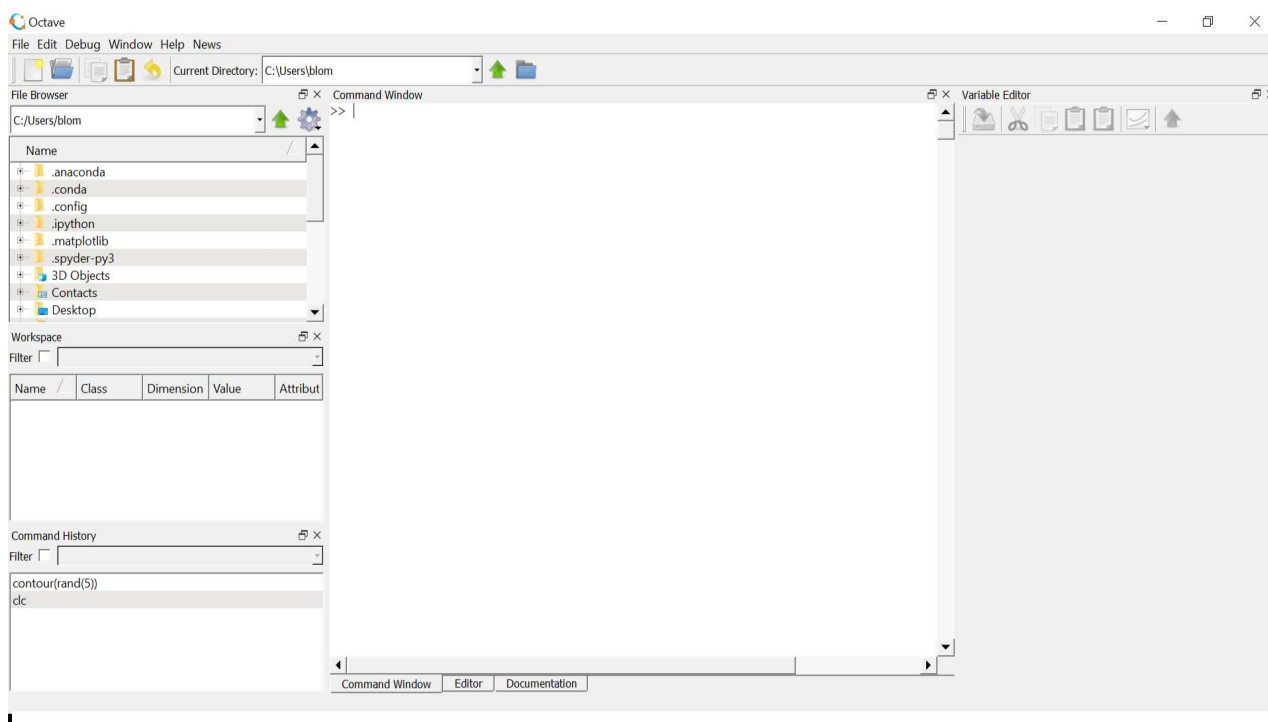
# Introduktion till OCTAVE

## 1 Inledning

OCTAVE är både en interaktiv matematikmiljö och ett programspråk, som används på många universitet och tekniska högskolor runt om i världen. Första versionen av OCTAVE skapades redan 1988 av John W. Eaton. Idag är OCTAVE främst känt för att vara ett av de stora fria alternativen till MATLAB.

## 2 Starta OCTAVE

När man startar OCTAVE ser det ut så här.



Det vi ser är OCTAVES Desktop.

De olika fönstren i Desktopen har namn (namnet står överst till vänster i fönstret). Det stora fönstret i mitten kallas **Command Window** och där kommer vi ge kommandon, till vänster ser vi ett lite mindre fönster som heter **Workspace** där vi ser vilka variabler vi har. Under **Workspace** finns ett fönster som heter **Command History** där vi ser vilka kommandon vi givit tidigare. Slutligen uppe till vänster ser vi **File Browser** som visar innehållet i aktuell mapp eller katalog.

### 3 En enkel beräkning och några grafer

Här följer några exempel så att vi snabbt kommer igång och ser lite resultat. Följ gärna med vid datorn och knappa in efter hand i **Command Window** och se vad som händer.

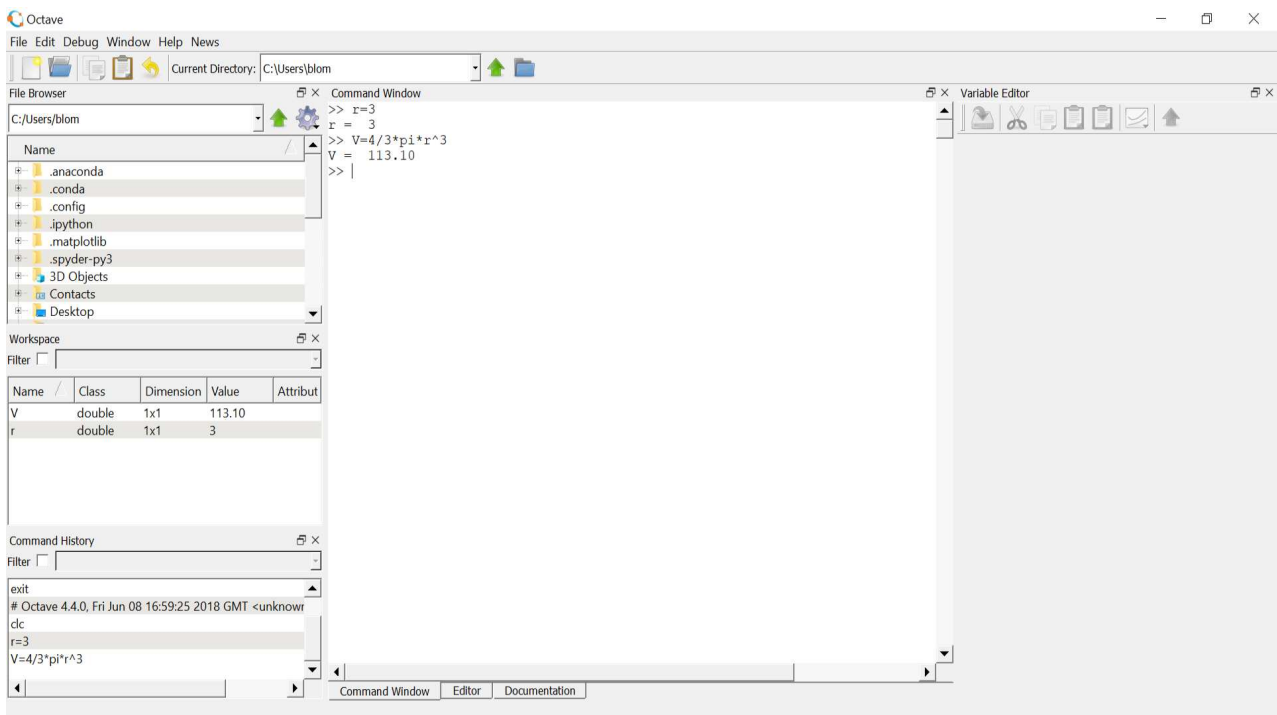
**Exempel 1.** Beräkna volymen av ett klot med radien  $r = 3$  cm. Volymen ges av  $V = \frac{4}{3}\pi r^3$ .

Först inför vi en variabel  $r$ , för radien, som vi ger värdet 3.

```
>> r=3
```

Därefter beräknar vi volymen enligt formeln ( $\pi$  ger en approximation av konstanten  $\pi$ ) och låter variabeln  $V$  få detta värde.

```
>> V=4/3*pi*r^3
```



Ett variabelnamn skall börja med en bokstav (**a-z, A-Z**), därefter får vi ha bokstäver (**a-z, A-Z**), siffror (**0-9**) och understrykningstecken (**\_**). OCTAVE skiljer på stora och små bokstäver.

Den s.k. promptern (**>>**) skriver vi inte. Tecknet finns i **Command Window** på raden där vi skall skriva vårt kommando och visar att OCTAVE är redo. Vi ser våra variabler och deras värden i **Workspace** och i **Command History** ser vi kommandona vi givit så långt.

**Uppgift 1.** Beräkna arean av en cirkelskiva med radien  $r = 4$  cm. Arean ges av  $A = \pi r^2$ .

**Exempel 2.** Rita grafen av  $f(x) = \sin(x) + 0.3 \sin(4x)$  för  $0 \leq x \leq 4\pi$ .

Först gör vi en lista eller radvektor  $x$  av  $x$ -värden mellan 0 och  $4\pi$ , med kommandot

```
>> x=0:0.1:4*pi;
```

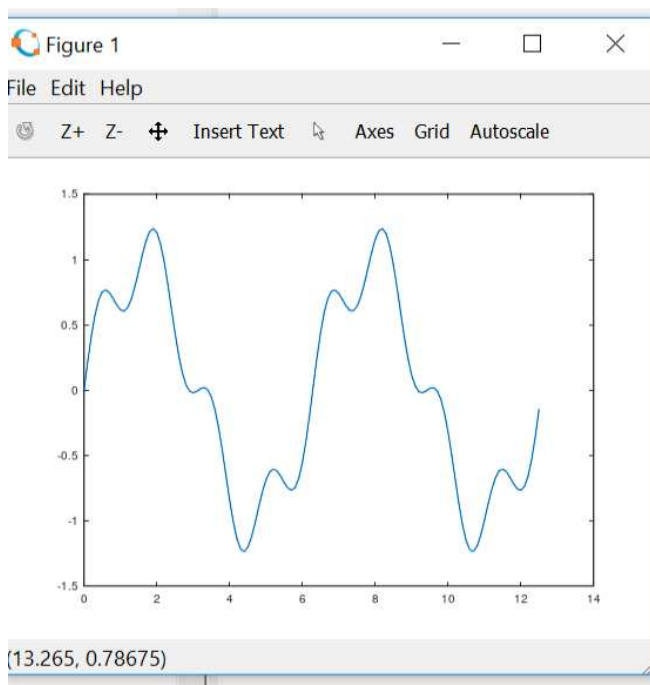
som vi skriver i **Command Window**.

Närmare bestämt får vi värdena 0, 0.1, 0.2, 0.3,  $\dots$ , 12.5, dvs. värden med start i 0, steget 0.1 och slut så nära upp mot  $4\pi$  som möjligt. Om vi hade inte skrivit ett semikolon (;) sist i uttrycket för  $x$ , hade alla  $x$ -värden skrivits ut i **Command Window**.

Därefter gör vi en lista eller radvektor  $f$  med  $f(x)$ -värden för varje  $x$ -värde i  $x$  och ritar upp grafen med `plot`.

```
>> f=sin(x)+0.3*sin(4*x);  
>> plot(x,f)
```

Dessa kommandon skriver vi i **Command Window** och ett grafikfönster **Figure** kommer upp



Vi kan använda uppåtpil ( $\uparrow$ ) för att komma till ett kommando vi givit tidigare, eller dubbelklicka på kommandot i **Command History**. Om vi vill kan vi gå längs raden med vänster- och högerpilarna ( $\leftarrow$ ), ( $\rightarrow$ ) och redigera kommandot. När kommandot ser ut som vi vill trycker vi på enter ( $\leftrightarrow$ ).

Vill vi rensa **Command Window** så ger vi kommandot `clc` och med kommandot `clf` rensar vi **Figure 1**.

**Uppgift 2.** Rita grafen till  $f(x) = \sin(x) + 0.3 \sin(5x)$  över intervallet  $0 \leq x \leq 4\pi$ .

**Exempel 3.** Rita graferna av  $f(x) = \sin(x)$  och  $g(x) = \sin(4x)$  för  $0 \leq x \leq 2\pi$ . Sätt rubrik och text på axlarna.

Vi använder funktionen `linspace` för att få 100 punkter jämnt fördelade mellan 0 och  $2\pi$ , då blir graferna jämna och snygga.

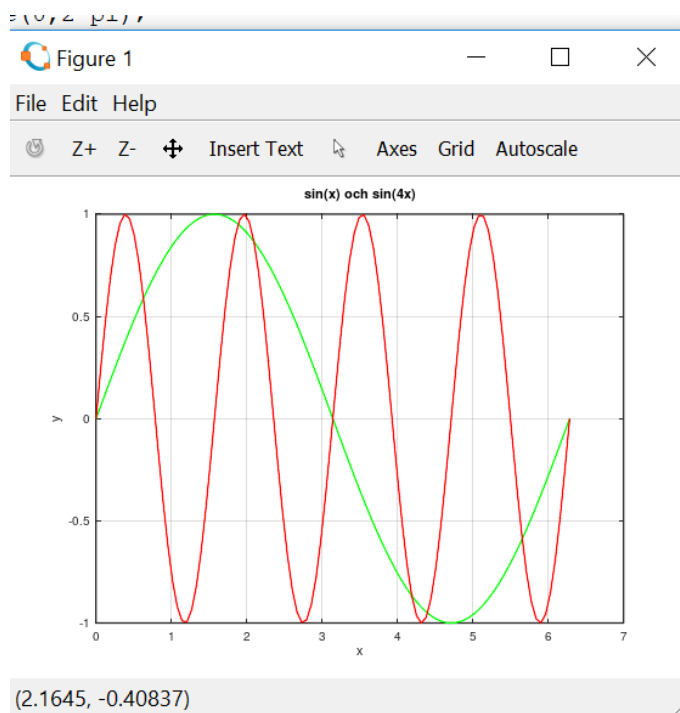
```
>> x=linspace(0,2*pi);  
>> f=sin(x);  
>> g=sin(4*x);
```

Vi ritar båda graferna samtidigt med `plot`, både paret `x, f` och paret `x, g`.

```
>> plot(x,f,'g',x,g,'r')
```

För att skilja graferna åt gjorde vi  $\sin(x)$ -grafens gröna `'g'` och  $\sin(4x)$ -grafens röda `'r'`. Vi sätter text på axlarna och rubrik samt lägger på ett rutnät med

```
>> xlabel('x'), ylabel('y')  
>> title('sin(x) och sin(4x)')  
>> grid on
```



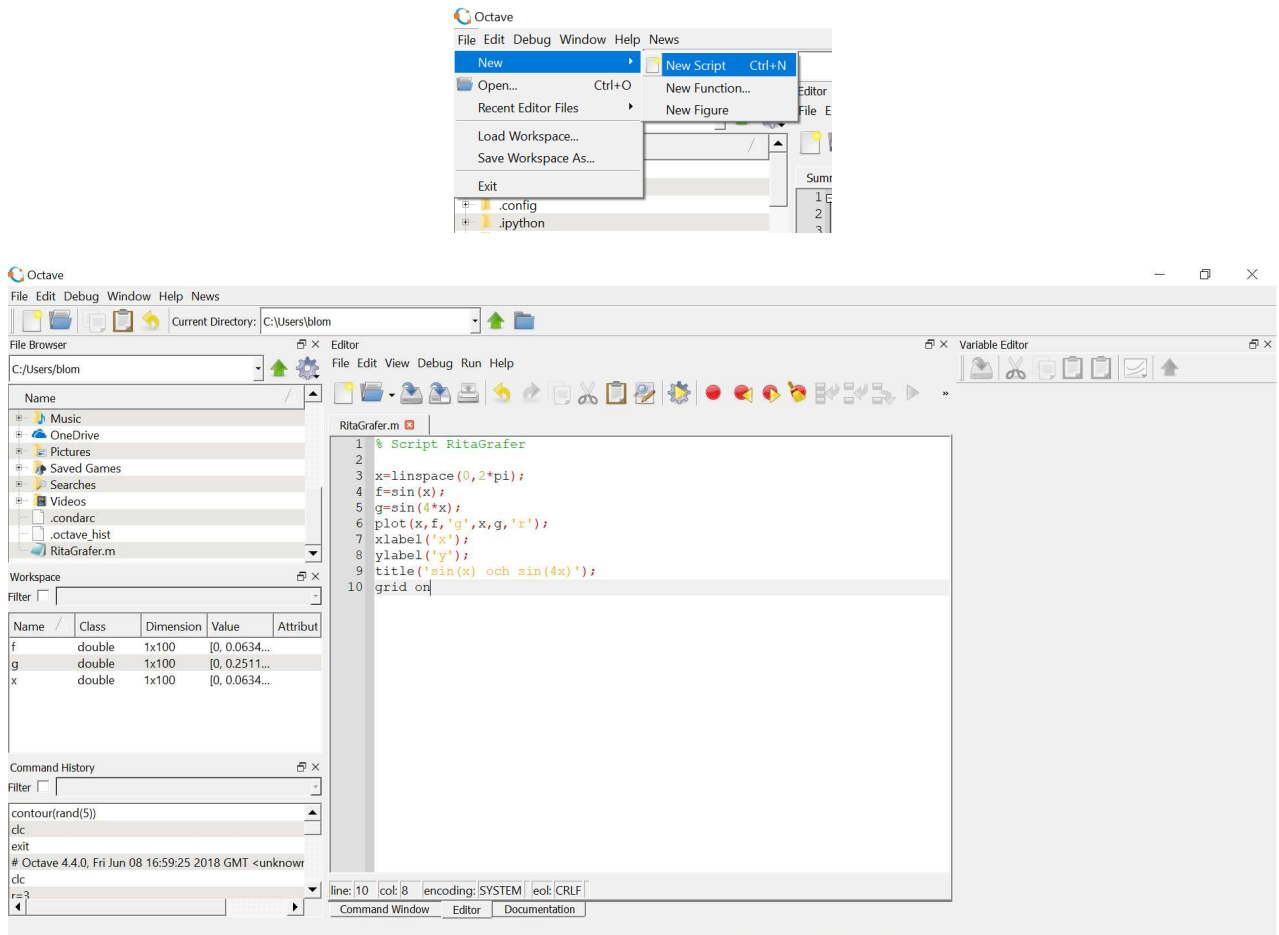
Texterna inom apostrofer (`' '`), t.ex. `'g'` och `'x'`, är s.k. textsträngar.

## 4 Script

För att slippa skriva om sina kommandon, eller bläddra med uppåt- och nedåtpilar ( $\uparrow$ ), ( $\downarrow$ ) i kommandofönstrets historik eller dubbelklicka i **Command History**, så brukar man skriva ett **script**. Ett **script** är en textfil som innehåller det man skulle kunna skriva direkt vid promptern (`>>`) i **Command Window**, och som utförs i OCTAVE då man ger textfilens namn som kommando.

Som exempel ser vi på ett **script** för exempel 3 gjort med den i OCTAVE inbyggda editorn.

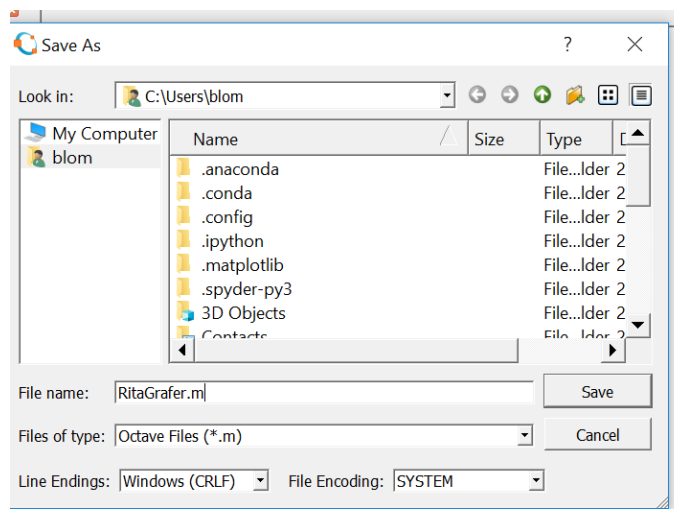
Editorn i OCTAVE startas genom att man trycker på fliken **Editor** under kommandofönstret. Man kan också öppna editorn genom att välja först **New** sedan **New Script** från **File**-menyn högst uppe till vänster på desktopen.



Editorn markerar koden med olika färger för att visa vad som är kommentarer, nyckelord, textsträngar, etc. (Kommentarer inleds med procenttecken.)

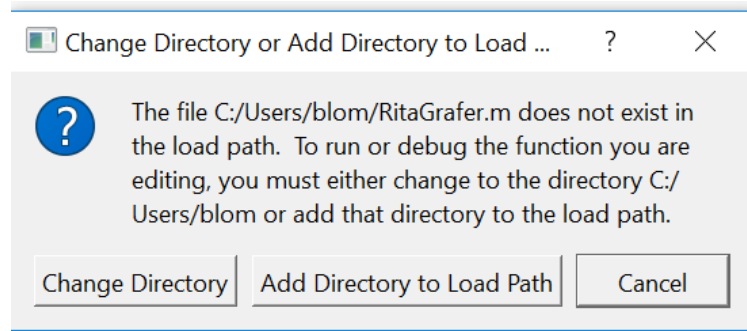
Spara kan vi göra genom att välja **Save File** från **File**-menyn eller trycka på tredje symbolen från vänster i editorn. Då sparas vårt script, under ett namn vi väljer, och utförs som om vi gav namnet som ett kommando. När scriptet körs kommer OCTAVE att utföra rad för rad (med start från första raden), och vi kommer få samma grafer som i exempel 3.

Så här ser dialogrutan ut som kommer upp då vi skall namnge vårt script.



Utanför OCTAVE får namnet på ett **script** tillägget `.m` för att skilja denna typ av fil från andra filer.

För att OCTAVE skall hitta filen, krävs det att katalogen där filen ligger är aktuell katalog. Om man försöker köra ett **script** som ligger i en annan katalog än den aktuella, så får man upp en fråga om att byta till den katalogen:



Välj **Change Directory** så byter OCTAVE katalog.

Aktuell katalog heter **Current Directory** i OCTAVE och visas i den avlånga rutan högst upp på desktopen. Man kan byta katalog genom att antingen klicka sig fram i **File Browser** eller genom att välja från de alternativ som finns i **Current Directory**-rutan.

## 5 Lite programmering

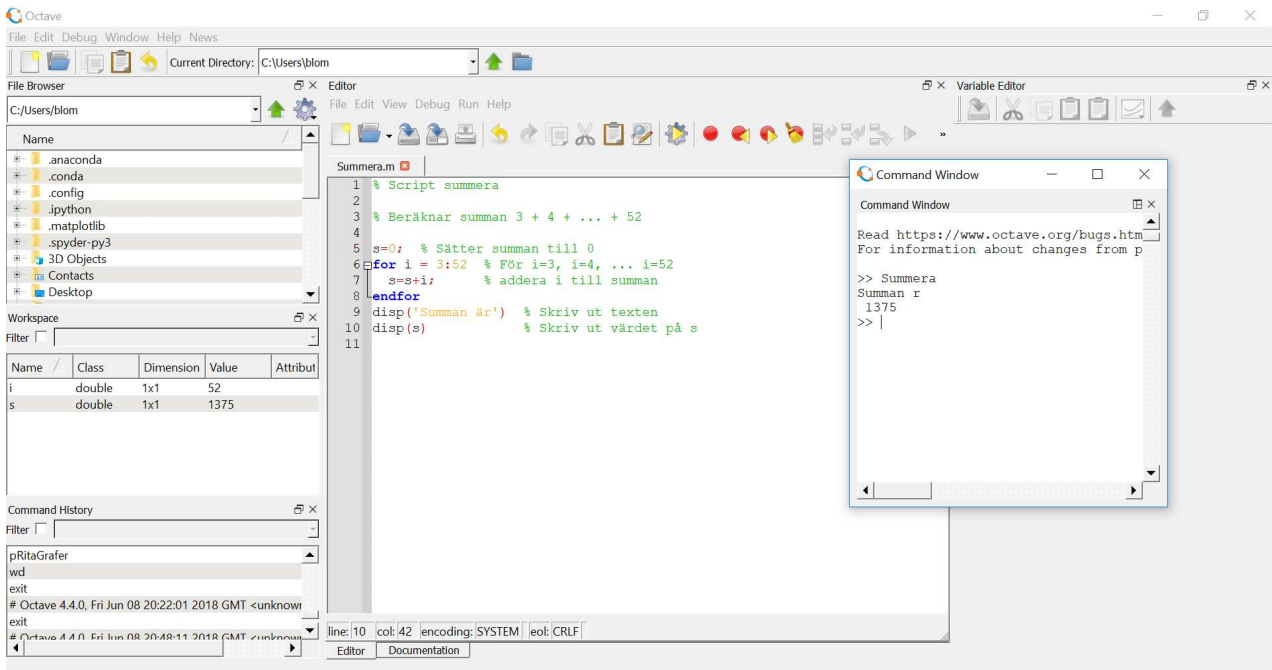
I OCTAVE finns repetitions- och villkorssatser. Vi nöjer oss för tillfället med att se på en repetitions-sats, en **for**-sats, som vi använder för att beräkna en summa i följande exempel.

**Exempel 4.** Beräkna summan  $s = 3 + 4 + 5 + \dots + 52$

Vi gör ett **script** med programkoden

```
s=0;
for i=3:52
    s=s+i;
endfor
```

Första satsen `s=0` är en tilldelningssats, variabeln `s` ges värdet 0. Andra satsen `for i=3:52` är en repetitionssats, den utför alla satser som följer ända ned till `end`, först för `i=3` sedan för `i=4` osv., ända tills satserna slutligen utförs för `i=52`. Första gången, dvs. då `i=3`, ges `s` ett nytt värde som är summan av gamla värdet på `s`, dvs. 0, och värdet på `i`, dvs. 3. Alltså kommer `s` ges värdet  $0+3=3$ . Andra gången, dvs. då `i=4`, ges `s` återigen ett nytt värde som är summan av gamla värdet på `s`, dvs. 3, och värdet på `i`, dvs. 4. Alltså kommer `s` ges värdet  $3+4=7$ . Så här fortsätter det ända tills `i=52` och `s` får sitt slutgiltiga värde.



Vi skriver lämpliga kommentarer (grön text) i programkoden och gör lämplig utskrift, först textsträngen `Summan` är och sedan summans värde.

På bilden har man lyft ut `Command Window`, det kan man göra genom att klicka på en liten symbol allra högst upp till höger i `Command Window`. (Alla fönster kan lyftas ut från desktopen på det här sättet). I exemplet körde man scriptet genom att skriva namnet i kommandofönstret.

I matematik skriver man gärna summan  $3 + 4 + 5 + \dots + 52$  med beteckningen

$$\sum_{i=3}^{52} i$$

**Uppgift 3.** Skriv ett script som beräknar summan

$$s = \sum_{i=1}^5 i^2 = 1^2 + 2^2 + 3^2 + 4^2 + 5^2$$

## 6 Function

Det finns olika sätt att göra egna funktioner i OCTAVE. Om funktionen innehåller flera uttryck eller satser måste man göra en `function`, dvs. skapa en textfil med funktionsbeskrivningen. Består funktionen av ett enda uttryck så kan vi göra en s.k. anonym funktion (`anonymous function`).

**Exempel 5.** Vi vill hitta ett nollställe till funktionen  $f(x) = x^3 - \cos(x)$ .

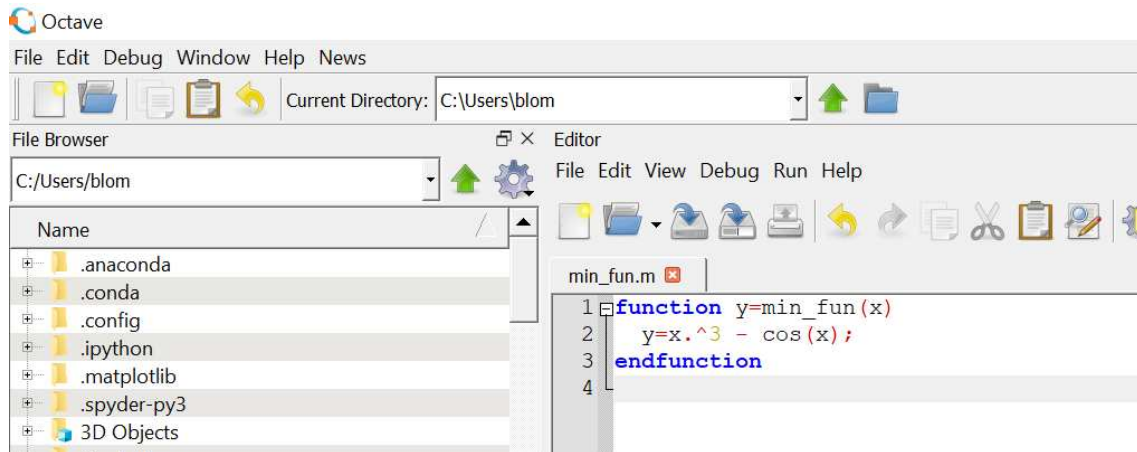
Det finns en funktion `fzero` i OCTAVE som hittar nollställena. För att använda `fzero` måste vi först beskriva vår funktion och det gör vi som en `function` enligt

```
function y=min_fun(x)
    y=x.^3-cos(x);
endfunction
```

där  $y$  är funktionens värde (utdata),  $x$  är funktionens argument (indata) och `min_fun` är funktionens namn (som vi själva valt).

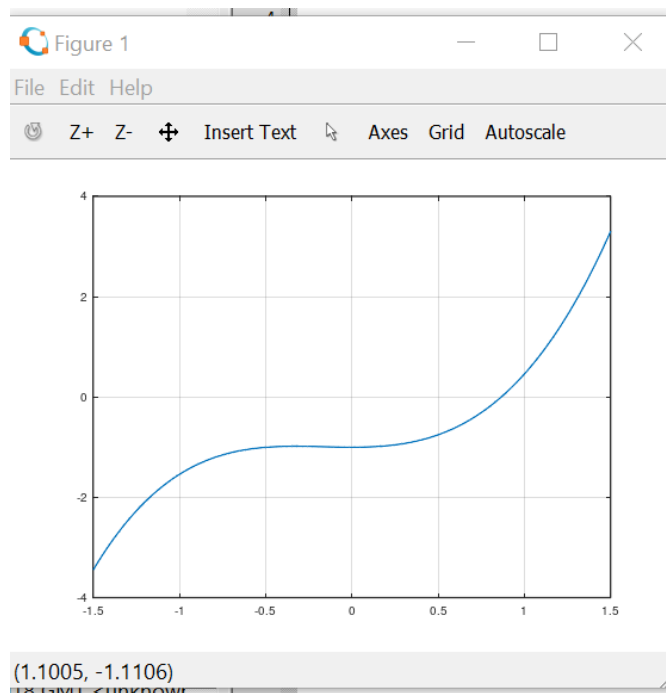
Vi skriver  $x^3$  som `x.^3` i OCTAVE eftersom vi vill att  $x$  skall kunna vara en lista eller radvektor med många  $x$ -värden och vill då att varje enskilt  $x$ -värde, dvs. varje element eller komponent, skall upphöjas till 3. Detta är en s.k. *komponentvis* operation.

Vi skriver in funktionen i editorn och sparar den under namnet `min_fun` på samma sätt som för ett `script`, dvs. textfilen skall heta `min_fun.m` i katalogen.



Vi ritar grafen genom att direkt i Command Window skriva

```
>> x=linspace(-1.5,1.5);
>> y=min_fun(x);
>> plot(x,y)
>> grid on
```



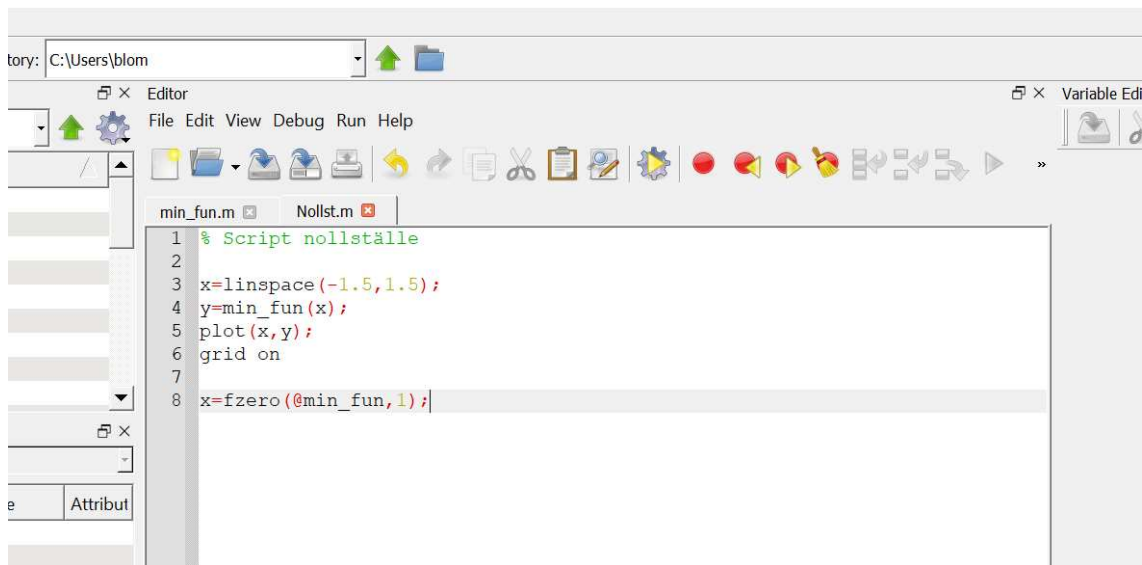
Vi ser att vi har ett nollställe nära  $x = 1$  och låter `fzero` beräkna nollstället noggrant med



```
>> z=fzero(@min_fun,1)
z =
    0.8655
```

Med `@min_fun` talar vi om för `fzero` vilken function som skall användas, dvs. vilken funktion det skall sökas nollställe till.

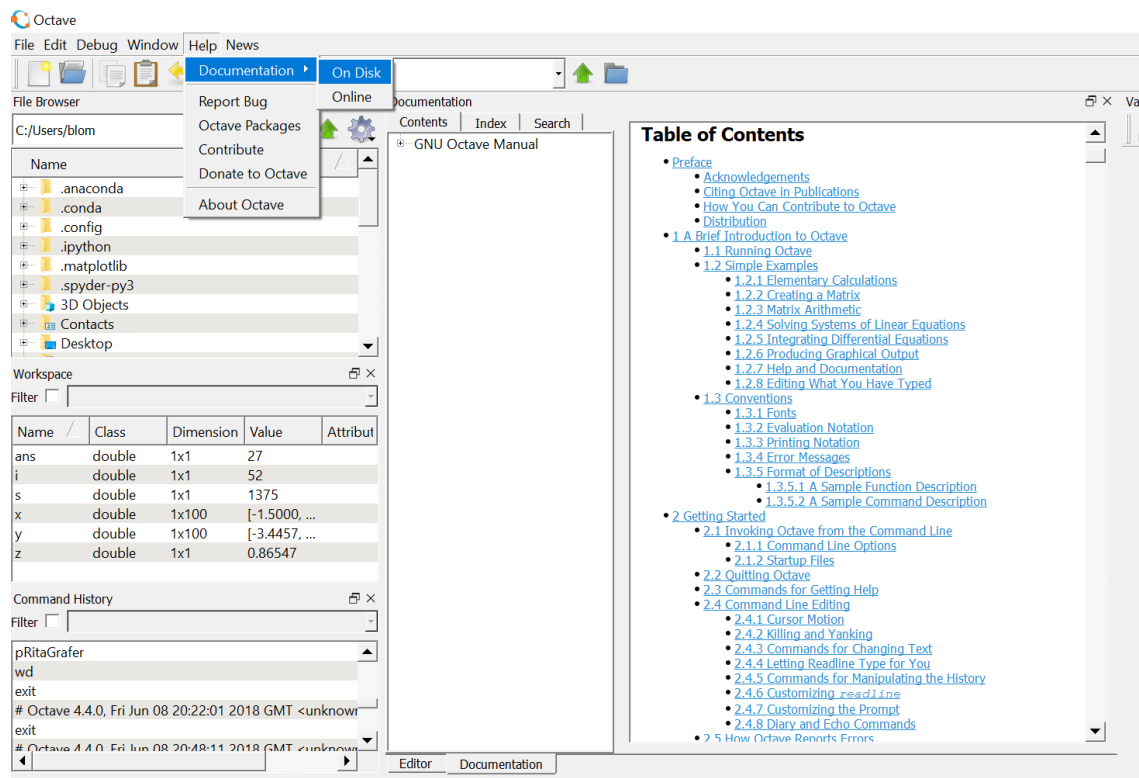
Vanligtvis kommer vi använda ett `script`.



**Uppgift 4.** Hitta alla nollställen till funktionen  $f(x) = x^2 - \cos(x)$ . Gör en function som beskriver vår funktion och rita en graf. Använd sedan `fzero` för att beräkna varje nollställe, ett i taget. Glöm inte att skriva  $x^2$  som `x.^2` i OCTAVE om `x` är en lista eller radvektor. Tänk på att funktionen måste skrivas i en egen textfil.

## 7 Help i OCTAVE

Det finns ett inbyggt hjälpsystem i OCTAVE. Välj alternativet `Documentation` följt av `On Disk` från `Help`-menyn (högst upp på desktopen). OCTAVE svarar med en beskrivning av själva programspråket OCTAVE och en beskrivning av hur desktopen används (Fönstret som innehåller rubriken `Table of Contents` till höger på bilden).

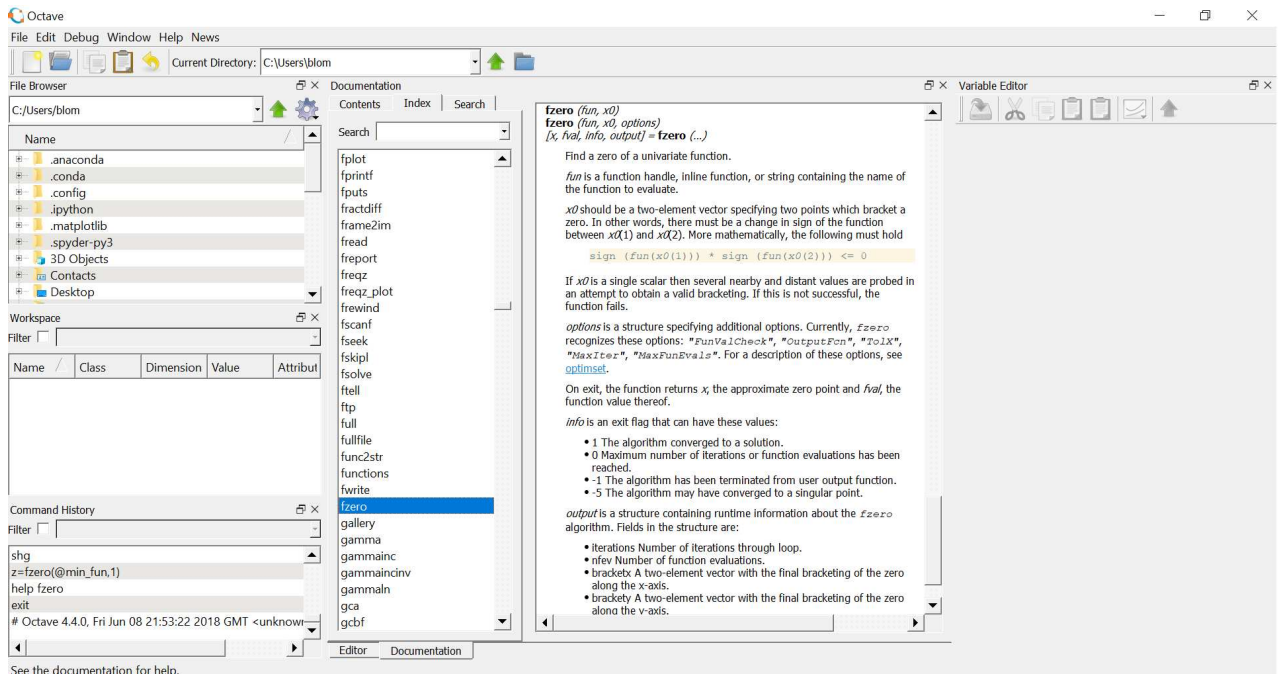


Det finns introduktionsavsnitt som ger vägledning för nybörjare – men hela hjälptexten är väldigt omfattande och får mer ses som en bok om OCTAVE.

Fönstret i mitten på bilden har tre flikar Contents, Index och Search. Om man klickar på Index får man en uppräknning av alla kommandon som finns att tillgå (de står i bokstavsordning).

Det är viktigt att lära sig att läsa dokumentationen. Den är inte skriven för att lära ut till nybörjare hur man löser ett problem med OCTAVE, utan för att visa exakt hur en funktion eller ett kommando används. Det är inte lättläst, och man måste lära sig att plocka fram den informationen som är av intresse för tillfället, dvs. man måste lära sig att ”skumma” texterna.

Vi letar upp `fzero` och dubbelklickar på namnet (det går också att skriva `fzero` i sökrutan). OCTAVE svarar då med en beskrivning av kommandot.



**Uppgift 5(a).** Leta upp oss läs hjälptexten för `linspace` som vi använde i samband med grafritning. Hur anger man antal punkter man vill ha? Hur många punkter får man som standard om man inte anger något antal?

**(b).** Leta själv upp hjälptexten för `fzero` (som vi ser ovan).