

The following is the code for evaluating a definite integral of a given function by a Numerical Method called Simpson's 3/8th Rule.

DOWNLOAD:[simpson2](#)

```
//Simpson's 3/8th Rule
//Evaluates the definite integral of a function f(x), from a to b.
//Written By: Manas Sharma(www.bragitoff.com)
funcprot(0);
function ans=simpson2(a,b,n,f)//function definition of simpson
    h=(b-a)/n;
    sum=0;
    for i=1:n-1
        x=a+i*h;
        if modulo(i,3)==0
            sum=sum+2*f(x);
        else
            sum=sum+3*f(x);
        end
    end
    ans=(3*h/8)*(f(a)+f(b)+sum);
endfunction
//NOTE: When the function is called the value of the third argument that is, n, should
be a multiple of 3.
```

You can either copy the code above and save it as a .sci file or download the file . Once you run the code, the function '*simpson2(a,b,n,f)*' can be called by other programs or even in the console.

*Function syntax:*

*simpson2(a,b,n,f)*

where,

*a=initial limit(real no.)*

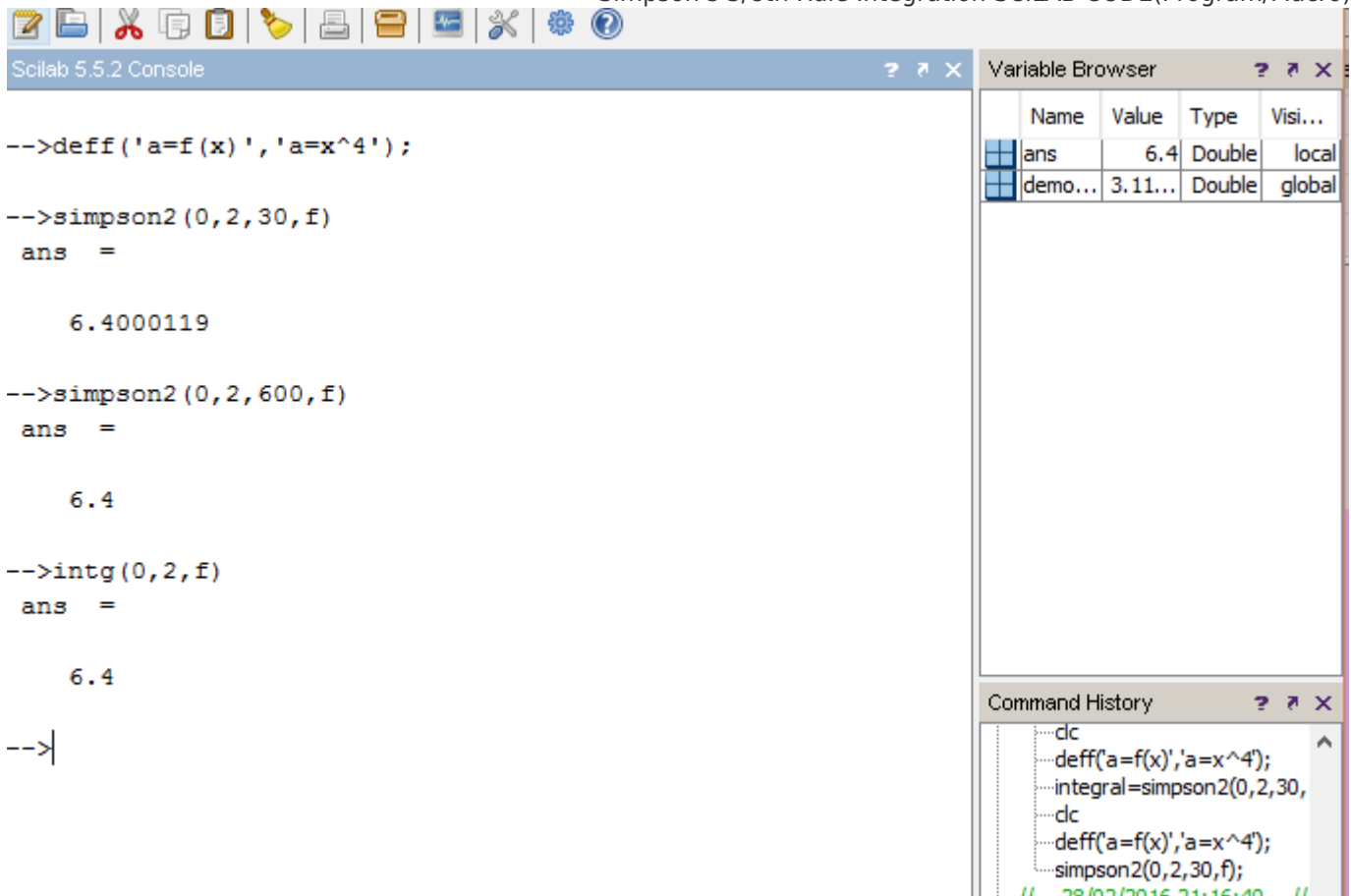
*b=final limit(real no.)*

*n=no. of sub-intervals(the higher the value of 'n' the better is the result.*

*NOTE: n should be a multiple of 3.*

Example:

The following code snippet evaluates the integral of  $1/(1+x^2)$  from 0 to 2.



The screenshot displays the Scilab 5.5.2 interface. The main console window shows the following commands and their outputs:

```
-->deff('a=f(x)', 'a=x^4');

-->simpson2(0,2,30,f)
ans =

    6.4000119

-->simpson2(0,2,600,f)
ans =

    6.4

-->intg(0,2,f)
ans =

    6.4

-->
```

The Variable Browser on the right shows the following variables:

| Name    | Value   | Type   | Visi... |
|---------|---------|--------|---------|
| ans     | 6.4     | Double | local   |
| demo... | 3.11... | Double | global  |

The Command History window at the bottom right shows the sequence of commands executed:

```
...clc
...deff('a=f(x)', 'a=x^4');
...integral=simpson2(0,2,30,
...clc
...deff('a=f(x)', 'a=x^4');
...simpson2(0,2,30,f);
...// -- 28/02/2016 21:16:40 -- //
```



## Manas Sharma

I'm a physicist specializing in computational material science with a PhD in Physics from Friedrich-Schiller University Jena, Germany. I write efficient codes for simulating light-matter interactions at atomic scales. I like to develop Physics, DFT, and Machine Learning related apps and software from time to time. Can code in most of the popular languages. I like to share my knowledge in Physics and applications using this Blog and a YouTube channel.

[manas.bragitoff.com/](http://manas.bragitoff.com/)









**Share this:**

[Click to share on Facebook \(Opens in new window\)](#)

[Click to share on Twitter \(Opens in new window\)](#)

[Click to share on WhatsApp \(Opens in new window\)](#)

[Click to share on Pinterest \(Opens in new window\)](#)

[Click to share on Reddit \(Opens in new window\)](#)

[Click to share on LinkedIn \(Opens in new window\)](#)

[Click to email a link to a friend \(Opens in new window\)](#)

[Click to print \(Opens in new window\)](#)

[Click to share on Tumblr \(Opens in new window\)](#)

[Click to share on Pocket \(Opens in new window\)](#)

[Click to share on Telegram \(Opens in new window\)](#)

[wpedon id="7041" align="center"]