

'eqnsolver' is a SCILAB toolbox that provides various functions to estimate the root of any equation by Numerical Methods.

The equation can be transcendental or algebraic.

It is available on the ATOMS: <https://atoms.scilab.org/toolboxes/eqnsolver>

There are 4 functions available for now, but I am still working on this toolbox so I may add a few more.

- nrsolve – Finds the real or complex root of an equation using a Numerical Technique called Newton-Raphson Method.
- sesolve – Finds the root of an equation using a Numerical Technique called Secant Method.
- bsolve – Finds the root of an equation within a given interval using the Bisection Method
- quadsolver – Finds the roots of a quadratic polynomial.

Demo:

Using the toolbox is pretty straight-forward. There are help pages for each of the function. You can load help by typing "help eqnsolver" in the Scilab console.

Suppose you want to find a root of the equation " $9x^3-81$ ". Then all you need to do is, first define a function:

```
deff('a=f(x)', 'a=9*x^3-81')
```

And then simply use the function, bsolve, sesolve as follows:

```
bsolve(0,5,f,0.0000001)
sesolve(5,25,f,0.0000001)
```

However to use the Newton-Raphson Method, you will also need to provide the derivative of the equation, so define a function 'df' which is the derivative 'f':

```
deff('a=df(x)', '27*x^2');
```

And then call the nrsolve() which will return a root.

```
nrsolve(5,f,df,0.0000001)
```

NOTE: The above functions never return all the root of the equation so you can adjust the initial guess that you give as arguments to get the different roots.

Details:

Toolbox: eqnsolver

Title: Equation Solvers (eqnsolver)

Summary: A toolbox with various functions for finding the roots of various algebraic or transcendental equations using Numerical Techniques.

Version: 1.0.0

Author: Manas Sharma

Maintainer: Manas Sharma

Category: Numerical Maths

License: Creative Commons Attribution-ShareAlike 3.0

ScilabVersion: ≥ 5.4

Author Notes:

A brand new [#scilab](#) Module created by me that lets you find the root of any algebraic or transcendental equation by Numerical Methods.

Numerical Methods are iterative methods that rely on an initial guess to find the exact root.

They only return a single root at a time and one needs to change the initial values for different roots.

One of the functions in the modules uses Newton-Raphson Method which is the best available function to find the roots available in the toolbox.

It can even find complex roots if the initial guess is complex.

Hope you like it!

The module is still a work in progress and I'm working on more functionalities to calculate the more roots simultaneously.

Anyone is welcome to contribute to this project.

Will keep you updated on future developments.



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/



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"]