

Screenshots Module 6: Root Solver

Alexander Ricciardi

Colorado State University Global

CSC505: Principles of Programming

Professor: Dr. Joseph Issa

December 21, 2025

Screenshots Module 6: Root Solver

This document contains a screenshot of the Root Solver script user interactions and outputs

Figure 1

Project Outputs on Terminal – User Flow

```

Windows PowerShell
PS P:\CSU-projects\CSU-505-Programs\CTA-Module-6> uv run python root_solver.py

Root Resolver

Main Menu
1. Solve a root
2. How to write a transcendental equation
3. Exit
Enter a choice (1, 2, or 3): 2

How to Write a Transcendental Equation

What is a transcendental equation?
An equation involving transcendental functions such as exp, log, sin, cos, tan, sqrt, or x^x.

Syntax tips:
- Use any 'x' as the variable.
- Use ^ or ** for exponents (e.g., x^3, x^(1/2), x^x).
- Use pow(x,y) for exponents (e.g., pow(x,3), pow(x,1/3)).
- Use log(x) for logarithmic functions.
- Use sin(x) for sine functions.
- Use cos(x) for cosine functions.
- Use tan(x) for tangent functions.
- Use abs(x) for absolute value functions.
- Use pi and e for mathematical constants.

Examples:
- cos(x)-x
- x^2-2
- sqrt(x)-1
- pow(x,3)-8
- x^(1/3)-2

Press Enter to continue...

Main Menu
1. Solve a root
2. How to write a transcendental equation
3. Exit
Enter a choice (1, 2, or 3): 1
Enter transcendental equation in x (e.g., cos(x)-x): cos(x)-x

Select Method
1. Bisection
2. Newton-Raphson
Enter a choice (1, or 2): 1

Bisection Method

Requires a sign change on [a, b]; repeatedly halves the interval.
You will enter:
- Tolerance (stopping threshold)
- Max iterations
- Interval start (a)
- Interval end (b)

Parameter notes:
- Tolerance: stop when |f(a)| <= tol or interval width <= tol.
- Max iterations: safety cap on loop count.
- Interval start/end: [a, b] must bracket a sign change (f(a)*f(b) < 0).

Enter tolerance (>0): 1e-6
Enter max iterations (>=1): 100
Enter interval start a: 0
Enter interval end b: 1

Equation: cos(x)-x
Method: Bisection
Iterations: 20
Root: 0.7390851974
Residual: 1.075e-07
Status: CONVERGED
Converged by tolerance.

Main Menu
1. Solve a root
2. How to write a transcendental equation
3. Exit
Enter a choice (1, 2, or 3): 1
Enter transcendental equation in x (e.g., cos(x)-x): x^3-x-2

Select Method
1. Bisection
2. Newton-Raphson
Enter a choice (1, or 2): 2

Newton-Raphson Method

Uses derivative to refine an initial guess x0.
You will enter:
- Tolerance (stopping threshold)
- Max iterations
- Initial guess x0
- Derivative step h (e.g., 1e-6)

Parameter notes:
- Tolerance: stop when residual or step size is below tol.
- Max iterations: safety cap on loop count.
- Initial guess x0: starting point; better guesses converge faster.
- Derivative step h: finite-difference step; too small may be unstable.

Enter tolerance (>0): 1e-6
Enter max iterations (>=1): 20
Enter initial guess x0: 1.5
Enter derivative step h (e.g., 1e-6): 1e-6

Equation: x^3-x-2
Method: Newton
Iterations: 3
Root: 1.5213798069
Residual: 5.899e-07
Status: CONVERGED
Converged by residual tolerance.

Main Menu
1. Solve a root
2. How to write a transcendental equation
3. Exit
Enter a choice (1, 2, or 3): 2
Are you sure you want to quit? [Y/N]: y
Goodbye!
PS P:\CSU-projects\CSU-505-Programs\CTA-Module-6>

```

Continue next page

Figure 2

Project Outputs on Terminal – Input Validation

```

Windows PowerShell
PS P:\CSU-projects\CSU-595-Programs\CTA-Module-6> uv run python root_solver.py

Root Resolver

Main Menu
1. Solve a root
2. How to write a transcendental equation
3. Exit

Enter a choice (1, 2, or 3): ww
Invalid input. Please enter a nonzero positive integer (e.g., 2).
Enter a choice (1, 2, or 3): -1
Invalid input. Please enter a nonzero positive integer (e.g., 2).
Enter a choice (1, 2, or 3): 4
Please select a valid option.
Enter a choice (1, 2, or 3): 2.2
Invalid input. Please enter a nonzero positive integer (e.g., 2).
Enter a choice (1, 2, or 3): 1

Enter transcendental equation in x (e.g., cos(x)-x): cos(y)-x
Invalid expression syntax: unmatched ')' (<unknown>, line 1)
Enter transcendental equation in x (e.g., cos(x)-x): cos(y)-x
Only variable 'x' and approved math names are allowed.
Enter transcendental equation in x (e.g., cos(x)-x): ccas(x)-tag(x)
Only approved math functions are allowed.
Enter transcendental equation in x (e.g., cos(x)-x): cos(x)-tag(x)
Only approved math functions are allowed.
Enter transcendental equation in x (e.g., cos(x)-x): cos(x)-tan(x)

Select Method
1. Bisection
2. Newton-Raphson

Enter a choice (1, or 2): 1

Bisection Method

Requires a sign change on [a, b]; repeatedly halves the interval.
You will enter:
- Tolerance (stopping threshold)
- Max iterations
- Interval start (a)
- Interval end (b)

Parameter notes:
- Tolerance: stop when |f(m)| <= tol or interval width <= tol.
- Max iterations: safety cap on loop count.
- Interval start/end: [a, b] must bracket a sign change (f(a)+f(b) < 0).

Enter tolerance (>0): -2
Invalid input. Please enter a nonzero positive float (e.g., 12.99).
Enter tolerance (>0): 3.3
Enter max iterations (>=1): 0
Invalid input. Please enter a nonzero positive integer (e.g., 2).
Enter max iterations (>=1): 2.2
Invalid input. Please enter a nonzero positive integer (e.g., 2).
Enter max iterations (>=1): 10
Enter interval start a: 1
Enter interval end b: -1
Require a < b.
Enter interval start a: 1
Enter interval end b: 100

Equation: cos(x)-tan(x)
Method: Bisection
Iterations: 1
Root: 50.5000000000
Residual: 7.337e-01
Status: CONVERGED
Converged by tolerance.

Main Menu
1. Solve a root
2. How to write a transcendental equation
3. Exit

Enter a choice (1, 2, or 3): 2

How to Write a Transcendental Equation

What is a transcendental equation?
An equation involving transcendental functions such as exp, log, sin, cos, tan, sqrt, or x^x.

Syntax tips:
- Use any 'x' as the variable.
- Use ^ or ** for exponents (e.g., x^3, x^(1/2), x*x).
- Use pow(x,y) for exponents (e.g., pow(x,3), pow(x,1/3)).
- Use log(x) for logarithmic functions.
- Use sin(x) for sine functions.
- Use cos(x) for cosine functions.
- Use tan(x) for tangent functions.
- Use abs(x) for absolute value functions.
- Use pi and e for mathematical constants.

Examples:
- cos(x)-x
- x*x-2
- sqrt(x)-1
- pow(x,3)-8
- x*(1/3)-2

Press Enter to continue...

Main Menu
1. Solve a root
2. How to write a transcendental equation
3. Exit

Enter a choice (1, 2, or 3): 1

Enter transcendental equation in x (e.g., cos(x)-x): cos(x)-x

Select Method
1. Bisection
2. Newton-Raphson

Enter a choice (1, or 2): 2

```

Continue next page

Newton-Raphson Method

Uses derivative to refine an initial guess x_0 .

You will enter:

- Tolerance (stopping threshold)
- Max iterations
- Initial guess x_0
- Derivative step h (e.g., $1e-6$)

Parameter notes:

- Tolerance: stop when residual or step size is below tol.
- Max iterations: safety cap on loop count.
- Initial guess x_0 : starting point; better guesses converge faster.
- Derivative step h : finite-difference step; too small may be unstable.

Enter tolerance (>0): 1

Enter max iterations (≥ 1): 100

Enter initial guess x_0 : tt

Invalid input. Please enter a valid float (e.g., 12.99).

Enter initial guess x_0 : -12.99

Enter derivative step h (e.g., $1e-6$): 1s-6

Invalid input. Please enter a nonzero positive float (e.g., 12.99).

Enter derivative step h (e.g., $1e-6$): 22e-10

Equation: $\cos(x)-x$

Method: Newton

Iterations: 40

Root: N/A

Residual: 2.198e+07

Status: FAILED_SAFELY

Derivative too small; failing safely.

Main Menu

1. Solve a root
2. How to write a transcendental equation
3. Exit

Enter a choice (1, 2, or 3): 3

Are you sure you want to quit? [Y/N]: 3

Invalid input. Please enter 'Y' or 'N'.

Are you sure you want to quit? [Y/N]: w

Invalid input. Please enter 'Y' or 'N'.

Are you sure you want to quit? [Y/N]: n

Main Menu

1. Solve a root
2. How to write a transcendental equation
3. Exit

Enter a choice (1, 2, or 3): 3

Are you sure you want to quit? [Y/N]: y

Goodbye!

PS P:\CSU-projects\CSC-505-Programs\CTA-Module-6> |