

**Peter Julius M. Estacio**

**Grouped with:**

**Augusto Gabbriel M. Calilung**

**Rafael Luis Mari R. Lambo**

**ENGG 27.01 - M**

**January 29, 2026**

## **Progress Report 1 for Project 1: Fixed-Point Approach for Finding Roots of Polynomials**

As of January 29, 2026, initial progress has been made with Augusto Gabbriel M. Calilung and Rafael Luis Mari R. Lambo as groupmates. Specifically, a C++ program has been developed to read a polynomial file, process the coefficients, utilize the fixed-point approach for all non-constant coefficients, and to determine whether the result is diverging or converging based on the iterations.

For file processing, a polynomial file called `test` was made using the basic example polynomial provided. The file is parsed and validated for the “POLYNOMIAL” token along with the degree of the polynomial, stored as a string and integer respectively. Using this stored integer, the program initializes an array of appropriate size and stores the coefficients indexed by their degree.

The user then is able to input a initial estimate  $g(0)$ . To implement the fixed-point approach, each non-constant term of the polynomial equation was isolated to form the function  $g(x) = x$ . Specifically, for any given non-constant term  $ax^n$ , the term would be transposed then the whole polynomial equation would be divided by  $ax^{(n-1)}$ , thus isolating  $x$  and representing  $g(x)$  as a rational function. For each iteration of  $g(x)$  then, the program attempts to use the fixed-point method to derive a root of  $f(x)$ .

Determining convergence or divergence was also initially implemented. Specifically, once the difference between the current iteration and the previous iteration equal 0, the program assumes convergence. The program then displays the result at maximum precision available for a double. On the other hand, the program assumes divergence should the results equal NaN, inf, or -inf. Currently, the fixed-point method is limited to 100 iterations should the problem face problems detecting convergence and divergence.

Further progress includes but is not limited to more robust convergence/divergence detection (especially in the case of oscillating results), header implementation, flexible polynomial file validation, and stricter user input validation. Screenshots of the initial results are shown below.

```

pedr0@pedr0:~/Desktop/Estacio_Pete:$ ./main
Enter filename: test
Header is present.
Read polynomial of index 3
=====
Initial Guess (g_0): 1
=====
Showing estimates for i = 3:
x_0 = 17.4
x_1 = -1.01723
x_2 = -1.95312
x_3 = -4.05105
x_4 = -3.39697
x_5 = -3.61145
x_6 = -3.5375
x_7 = -3.56262
x_8 = -3.55404
x_9 = -3.55696
x_10 = -3.55597
x_11 = -3.55631
x_12 = -3.55619
x_13 = -3.55623
x_14 = -3.55622
x_15 = -3.55622
x_16 = -3.55622
x_17 = -3.55622
x_18 = -3.55622
x_19 = -3.55622
x_20 = -3.55622
x_21 = -3.55622
x_22 = -3.55622
x_23 = -3.55622
x_24 = -3.55622
x_25 = -3.55622
x_26 = -3.55622
x_27 = -3.55622
x_28 = -3.55622
x_29 = -3.55622
x_30 = -3.55622
x_31 = -3.55622
x_32 = -3.55622
x_33 = -3.55622
x_34 = -3.55622
x_35 = -3.55622
x_36 = -3.55622
x_37 = -3.55622
x_38 = -3.55622
Converged. Final result: -3.5562190082267491
=====

=====
Showing estimates for i = 2:
x_0 = 11.25
x_1 = -72.5793
x_2 = -3286.43
x_3 = -6.75039e+06
x_4 = -2.84798e+13
x_5 = -5.06938e+26
x_6 = -1.60617e+53
x_7 = -1.61235e+106
x_8 = -inf
Diverged. Iterations stopped.
=====

=====
Showing estimates for i = 1:
x_0 = -0.708333
x_1 = -0.932564
x_2 = -0.918703
x_3 = -0.919268
x_4 = -0.919244
x_5 = -0.919245
x_6 = -0.919245
x_7 = -0.919245
x_8 = -0.919245
x_9 = -0.919245
x_10 = -0.919245
x_11 = -0.919245
x_12 = -0.919245
x_13 = -0.919245
Converged. Final result: -0.91924522833031541
=====
```

sample screenshots – program executable run and example results

```

* Executing task: C/C++: LINUX BUILD (g++)
Starting build...
/usr/bin/g++ -fdiagnostics-color=always -g *.cpp -o '/home/pedr0/Documents/Coding/ENGG 27 - 2/Project 1/main'
Build finished successfully.
* Terminal will be reused by tasks, press any key to close it.
```

sample screenshot – compilation on Visual Studio Code

```

● pedr0@pedr0:~/Documents/Coding$ g++ --version
g++ (Debian 12.2.0-14+deb12u1) 12.2.0
Copyright (C) 2022 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

compiler version – g++ (Debian 12.2.0-14+deb12u1) 12.2.0

Operating System	Debian GNU/Linux 12 bookworm (x86-64)
Cinnamon Version	5.6.8
Linux Kernel	6.1.0-39-amd64
Processor	Intel® Core™ i5-8250U CPU @ 1.60GHz × 4
Memory	15.5 GiB
Hard Drives	628.6 GB
Graphics Card	Intel Corporation UHD Graphics 620
Upload system information	
Copy to clipboard	

operating system – Debian 12 bookworm, Cinnamon version 5.6.8

```
{ + 1; i++)      Visual Studio Code

Version: 1.104.0
Commit: f220831ea2d946c0dcb0f3eaa480eb435a2c1260
Date: 2025-09-10T06:46:18.035Z (4 mos ago)
Electron: 37.3.1
ElectronBuildId: 12342881
Chromium: 138.0.7204.235
Node.js: 22.18.0
V8: 13.8.258.31-electron.0
OS: Linux x64 6.1.0-39-amd64
}, i--)
```

IDE – Visual Studio Code ver 1.104.0