Implementation and Improvement of the Continuous Newton Method

Yicheng Zhang

Dept. of Electronic and Electrical Engineering
Southern University of Science and Technology
Shenzhen, China
zhangyc2022@mail.sustech.edu.cn

Abstract—**This project implements and evaluates the Continuous Newton-Raphson method (CNR), and improves it with momentum method to avoid stagnation at local minima.**

Index Terms—**Adaptive step size control, Continuous Newton–Raphson method, Momentum method, Newton–Raphson methods, Nonlinear differential equations**

I. INTRODUCTION

This project implements the Continuous Newton-Raphson method (CNR), explores its performance, and improves it with momentum method. CNR is a method to improve the traditional Newton-Raphson method (NR) by treating the original problem as a differential equations problem and introducing adaptive step size control which produces more conservative step sizes. CNR method shows better performance than NR, providing more stable convergence, and, with appropriate parameter, a faster convergence rate. CNR, however, is sometimes so conservative that it may stagnates at local minima, and this project improves it with the momentum method.

II. PAPER REVIEW

The paper by Amrein[1] brought out a method to improve the vanilla Newton Raphson method. To solve the problem

$$x \in \Omega: \quad \mathbf{F}(x) = 0 \tag{1}$$

for suffiently small t_n, Newton-Raphson method is written in the following way

$$x_{n+1} = x_n - t_n \mathbf{F}'(x_n)^{-1} \mathbf{F}(x_n) \tag{2}$$

Define $\mathbf{N_F} = -\mathbf{F}'(x)^{-1} \mathbf{F}(x_n)$, the above problem can be considered as the discretization of the differential equation problem

$$\begin{cases} \dot{x}(t) = \mathbf{N_F}(x(t)), & t \geq 0 \\ x(0) = x_0 \end{cases} \tag{3}$$

Define the set of all points which belong to trajectories leading to x_∞, the desired solution, as $\mathcal{A}(x_\infty)$, which is an attractor. If the iterations stay within the same attractor, the chaotic behavior of Newton method will be tamed. This also explains one reason of why vanilla Newton method fails: step size is too large that the iteration falls outside of the attractor $\mathcal{A}(x_\infty)$.

Consider the linearization at $t = 0, x(0) = x_0$, i.e.

$$\hat{x}(t) = x_0 + t\dot{x}(0) \tag{4}$$

The paper shows that

$$\hat{x}(t) - x(t) = \frac{1}{2}t^2 \|\mathbf{N_F}(x_0)\|_X \\ + O(t^3) + O(\|x(t) - x_0\|_X^2) \tag{5}$$

Thence, fixing a tolerance $\tau > 0$ such that

$$\tau = \|\hat{x}(t) - x(t)\|_X \\ = \frac{t^2}{2}\|\mathbf{N_F}(x_0)\|_X + O(t^3) + O(\|x(t) - x_0\|_X^2) \tag{6}$$

gives the following adaptive step size control for the Newton method:

1) Start the Newton iteration with an initial guess $x_0 \in \mathcal{A}(x_\infty)$.
2) In each iteration step $n = 0, 1, 2, ...$, compute

$$t_n = \min\left(\sqrt{\frac{2\tau}{\|\mathbf{N_F}(x_n)\|_X}}, 1\right) \tag{7}$$

3) Compute x_{n+1} based on the Newton iteration, and go to the next step.

It shall be noted that tolerance τ is set manually a priori. Smaller τ will lead to more conservative step sizes. Larger τ usually means larger step sizes, but due to the min function, it will not exceed vanilla NR method.

III. DEMONSTRATION OF CNR METHOD

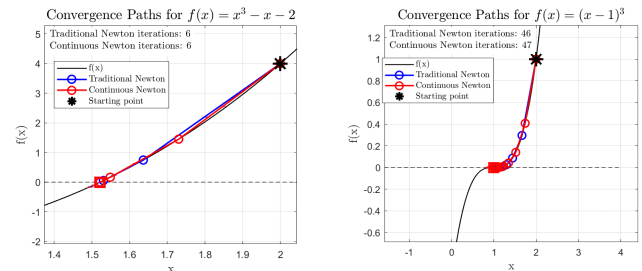### A. NR already wins: CNR preserves quadratic convergence

Fig. 1: Simple cases where both NR and CNR succeed

Both NR and CNR succeed, and CNR looks slightly more conservative, yet not much slower. As shown in the figure,

the min function helps preserve the quadratic convergence of NR.

B. NR explodes/converges to wrong solutions: CNR can avoid it
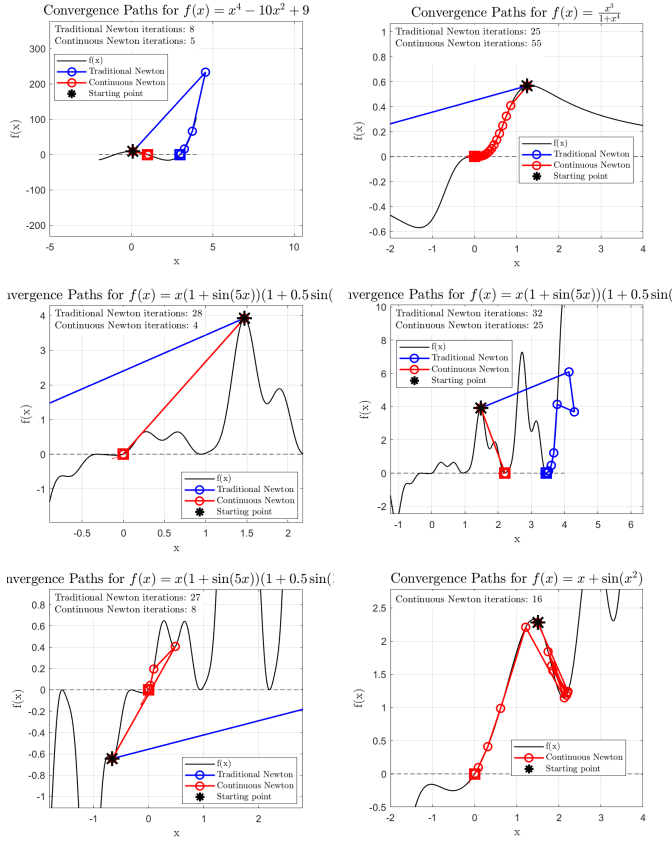

Fig. 2: NR explodes/converges to wrong solutions: CNR can avoid it
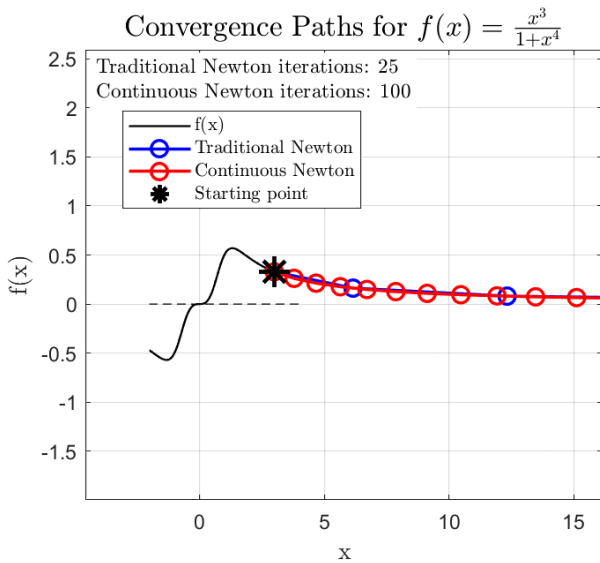
C. CNR doesn't save you from bad initial guess

Fig. 3: NR and CNR fail at bad initial guess

In this case, the initial guess is so bad that any Newton method will fail, because Newton update always has the wrong direction.
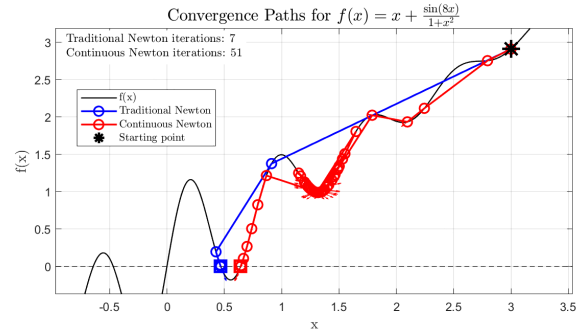
D. CNR stagnates at local minima

Fig. 4: CNR stagnates at local minima. Note that NR doesn't stagnate but converges to the wrong solution.

Recall that smaller τ leads to more conservative step sizes. In this case, τ is too small that CNR stagnates at the local minima.
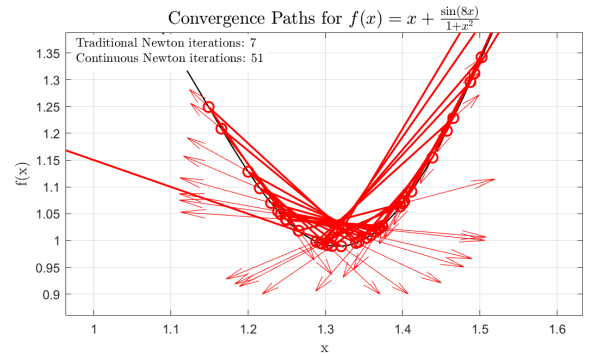

Fig. 5: CNR swings at local minima.

Observe it carefully, we note that CNR swings at the local minumum. This is expected because at this local minimum, $\|\mathbf{F}\|_X$ is much larger than 0, making the step size large enough that it won't "quadratically converge" to the basin.

When do CNR leaves the local minimum? When CNR accidentally steps at the bottom of the basin, the Newton update will be extremely large, and CNR will leave the local minimum. This stagnation produces an unwanted swinging behavior, which doesn't make CNR fail but causes it to be extremely slow.

IV. SOLUTION TO STAGNATION AT LOCAL MINIMA

A wrong solution to the stagnation is trying to make the step size larger, because it is hard to predict the precise dilation, where wrong dilation will produce an overshoot.

The idea of my solution is to make it converge to the local minimum as fast as possible. And then, since the bottom of the basin will produce a extremely large Newton update, the iteration will leave the local minimum.

To make sure this mechanism don't interfere with other nice properties of CNR, this mechanism is only triggered when swinging is detected. This is achieved by storing the last 4 iterations' Newton updates. If the last updates have alternating signs (for vectors, signs of dot product may be used), it is detected as swinging. Then, the new step size will be:

$$\Delta x'_{n+1} = \beta \Delta x_n + \Delta x_{n+1} \quad (8)$$

where β is a momentum coefficient, and is empirically set to 0.9. For example, if Δx_{n+1} is leftwards, and Δx_n is rightwards, then $\Delta x'_{n+1}$ will be offset by Δx_n so that it is only slightly leftwards. Then, overshoot is avoided so that CNR can converge to the local minimum.
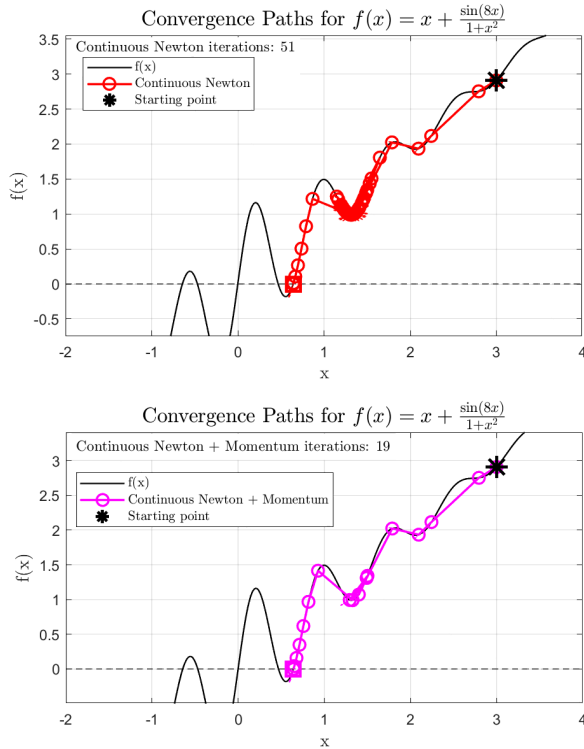

Fig. 6: CNR without/with momentum method

V. DISADVANTAGES SHOWN IN CIRCUITS SIMULATION

Though the CNR method looks promising, my tests on circuits simulation don't show much difference compared with vanilla NR method. I tested CNR on the validation tests in homeworks, and the results shows that:
1) If the tests are expected to fail, CNR method will fail as well.
2) If the tests are expected to succeed (whether with INIT/ LIMITING), CNR method will also succeed, but with a slightly slower convergence rate.

Note that I didn't made many tests. It may be possible that CNR method works better in some other cases. I think it's possible that the expected failure tests you devised are so poorly conditioned that CNR cannot make a difference.

However, the tests also show a disadvantage of CNR method: the manual setting of τ is crucial, and is hard to set. For example, in HW3/test_nrwil.test_vsrc_rdiode, NR with

INIT/LIMITING can solve, and CNR with $\tau = 100$ can also solve. However, if τ is smaller (e.g. 1), CNR will fail by exceeding max_iter because its step size is too small. This indicates that CNR method lacks a correction strategy for the predicted step size. Here, setting $\tau = 100$ wins because $\|\mathbf{N_F}(x_0)\|_X \approx 100$ in most of the time, making t look reasonable ($0.1 \sim 0.9$). However, I haven't found a good way to adjust τ adaptively, because $\|\mathbf{N_F}(x_0)\|_X$ can vary a lot, and changing τ will change the behavior of CNR.

VI. FUTURE WORK

1) Test CNR on more cases.
2) Improve the selection of τ. Possibly adjust it adaptively regarding $\|\mathbf{N_F}(x_0)\|_X$.
3) Draw more plots to help understand the behavior of CNR method, possibly with Newton fractal plots [2].
4) Check other methods to cope with stagnation at local minima. e.g., Backtracking Line Search, Trust Region Method.

VII. TAKEAWAYS

1) CNR method does, in many cases, avoid explosion or converging to wrong solutions.
2) (In my tests,) In circuits simulation, CNR method shows little improvement. INIT/LIMITING plays a more decisive role.
3) The manual setting of τ is crucial, but how to adjust it is still a problem.

REFERENCES

[1] M. Amrein and T. P. Wihler, "An adaptive Newton-method based on a dynamical systems approach," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 9, pp. 2958–2973, Sep. 2014, doi: 10.1016/j.cnsns.2014.02.010.
[2] 3Blue1Brown, "Newton's Fractal (which Newton knew nothing about)." 2021.