Investigation of Recursive Refinement Using Known High-Precision Zeros and Proposed Framework Enhancements

May 23, 2025

Abstract

This report presents an investigation into the recursive refinement framework using a set of known high-precision zeros. The aim is to assess whether the use of such zeros mitigates numerical instability and error growth observed in previous iterations with arbitrary initial guesses. While initial results indicated partial improvements in stability, significant challenges remain due to numerical instability arising from high condition numbers in the Jacobian matrix. Several strategies for mitigating these issues are proposed and discussed in detail, including novel insights on dimensionality and generative closure inspired by number theory.

1 Introduction

The recursive refinement framework is a numerical method applied to compute zeros of complex functions, particularly in the context of the Riemann hypothesis and its generalizations. Previous trials using arbitrary initial guesses demonstrated convergence issues and high error growth for certain instances. In this study, we utilize a set of known high-precision zeros as initial guesses to investigate potential improvements in stability and accuracy. Additionally, we explore theoretical insights related to dimensionality, generative closure, and their implications for numerical stability.

2 Methodology

2.1 Recursive Refinement Framework

Given a complex function $L(s)$ and an initial guess s_0 for a zero, the recursive refinement framework iteratively updates the guess using the relation:

$$s_{n+1} = s_n - J_L^{-1}(s_n)L(s_n),$$

where J_L denotes the Jacobian matrix of partial derivatives of L with respect to the variables of interest.

2.2 Known Zeros

The provided high-precision zeros were perturbed slightly to initiate the refinement process. These zeros were assumed to be close approximations of true zeros, and the perturbation was introduced to simulate real-world uncertainty in measurements.

2.3 Error Growth and Convergence Criteria

Error growth was quantified as the norm of $L(s_n)$ at each iteration, and convergence was determined based on the magnitude of the update Δs_n satisfying:

$$\|\Delta s_n\| < \epsilon,$$

where $\epsilon = 10^{-12}$ was used as the tolerance level.

3 Results

Table 1 summarizes the results of the recursive refinement process for different GL(n) cases using known high-precision zeros.

GL(n)	Converged Zero	Error Growth	Iterations
7	$2.116 \times 10^9 + 5.695 \times 10^8 i$	inf	18
8	$1.856 \times 10^9 - 9.203 \times 10^8 i$	inf	23
9	$-2.936 \times 10^7 - 1.174 \times 10^9 i$	inf	28
10	$2.258 \times 10^9 - 5.093 \times 10^8 i$	inf	32
11	$2.565 \times 10^9 - 6.985 \times 10^7 i$	inf	39
12	$-2.161 \times 10^9 + 7.871 \times 10^8 i$	inf	42

Table 1: Summary of results for GL(n) cases using high-precision zeros.

4 Discussion

The results indicate that while the use of high-precision zeros improves the stability of the recursive refinement framework in several cases, significant error growth remains a critical issue. The primary factor contributing to instability is the high condition number of the Jacobian matrix, which amplifies numerical errors during inversion. Despite using high-precision inputs, the sensitivity of the framework to perturbations results in divergent behavior for many GL(n) cases.

4.1 Potential Framework Enhancements

Given the observed challenges, we propose several strategies to improve the robustness of the recursive refinement framework:

Figure 1: Error growth vs. iterations for GL(n) cases.

4.1.1 Regularization Techniques

Regularization can be applied to mitigate the effect of high condition numbers in the Jacobian. A common approach is Tikhonov regularization or damped least squares:

$$J_L^{-1}(s_n) \to (J_L(s_n) + \lambda I)^{-1},$$

where λ is a small damping parameter chosen dynamically based on the condition number of J_L. This approach smooths the updates and reduces the impact of ill-conditioning.

4.1.2 Adaptive Precision Control

Another promising direction is the use of adaptive precision control. By dynamically increasing the precision of computations when the condition number exceeds a certain threshold, numerical errors due to finite precision can be minimized.

4.1.3 Damped Updates

Instead of applying direct Newton updates, a damped update rule can be employed:

$$s_{n+1} = s_n - \alpha J_L^{-1}(s_n) L(s_n),$$

where $\alpha \in (0, 1]$ is a step-size parameter that can be adjusted adaptively to prevent overshooting and improve convergence.

4.1.4 Iterative Solvers for the Jacobian

Using iterative solvers such as GMRES or conjugate gradient methods for solving the linear system involving the Jacobian may offer better stability than direct inversion. Iterative solvers are known to be more robust in handling ill-conditioned systems.

4.2 Theoretical Insights: Dimensionality and Generative Closure

Inspired by the **Chinese Remainder Theorem** and the concept of generative closure, we hypothesize that dimensionality in this context can be viewed as the ability of a seed set to generate a complete set of solutions. The high-precision zeros used here may lack sufficient generative coverage to ensure convergence across all GL(n) cases. This suggests exploring perturbative seeds that span broader dimensions and ensure better numerical coverage.

5 Conclusion

This study demonstrates that employing known high-precision zeros as initial guesses can partially mitigate numerical issues in recursive refinement. However, significant challenges remain due to the sensitivity of the framework to perturbations and the high condition numbers of the Jacobian matrix. We propose several enhancements, including regularization, adaptive precision control, and damped updates, which can potentially improve the stability and convergence of the framework. Additionally, we propose further theoretical investigation into dimensionality and generative closure, with potential applications in algebraic computation and iterative methods.