Program Structure and Development

Mingzheng Michael Huang

Translating Algorithm Components

- Repetition: Utilize ***while*** loops for conditions requiring repetitive action until met (e.g., reaching 50 iterations). Employ ***break*** statements to halt repetition for specific conditions (e.g., method convergence).
- Decision Making: Use ***if/else*** statements to handle decision points in the algorithm (e.g., handling singular matrices).
- Math: Directly translate algorithmic mathematical computations as code expressions (e.g., updating roots via Fixed-Point methods).
- Names: Map algorithmic variables to actual code variables with appropriate type definitions (e.g., representing the Jacobian matrix as a 2x2 array).
- Altering Values: Algorithm steps that manipulate values will become assignment statements (e.g., updating the root value with ***x = newX;***).
- Complicated Steps: Abstract complex algorithmic steps into separate functions for future definition (e.g., isolate the calculation and determinant of the Jacobian matrix to a separate function).
- And the answer is...!: When the algorithm concludes and derives the answer, write a ***return*** statement (e.g., returning a Boolean value in the ***has_converged*** function).

Program Validation

Stress Testing Approach (See README.TXT)

- Leveraged known solutions ($\pm\sqrt{3}/2$, $\pm 1/2$) to conduct rigorous stress tests for both Newton's Method and Fixed-Point Iteration.
- Employed a comprehensive sweep of initial guesses (x, y) to ascertain method robustness.

Validation Criteria

- For each method and initial guess, calculated (x, y) must align with one of the known points within a tolerance of 1e-10.
- Special cases considered: singular Jacobian in Newton's method and square root of negative numbers in Fixed-Point Iteration.
- The iteration limit is capped at 50 for both methods.

Results

- All tests successfully passed, corroborating that the solutions computed are in close agreement with the known analytical solutions within the set tolerance.

Challenges and Improvements

Most Challenging Aspects

- Algorithm Robustness: Ensuring the reliability of Newton's method and Fixed-point iteration for a broad range of initial guesses.
- Debugging Complexity: Bugs that arose during testing were hard to fix using only paper and pencil without specialized debugging tools.

Key Improvements in Final Submission

- Stress Testing: A comprehensive stress test suite was added to validate both algorithms, offering proof of their robustness across a broad range of initial guesses.
- Debugging: Initial bugs were identified and fixed using debugging tools like Valgrind and gdb.