Verification and Validation Report: MPIR

Xunzhou (Joe) Ye

April 16, 2025

1 Revision History

Date	Version	Notes
16 April 2025	1.0	Initial draft

2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test

Contents

List of Tables

List of Figures

3 Functional Requirements Evaluation

In this section, the system tests that will be conducted are described in detail. These tests will be used to verify the fulfillment of the functional requirements as listed in the SRS (Ye 2025).

3.0.1 Matrix Inputs and Outputs

This section covers the requirement R3 of the SRS. This includes essentially a "driver" for the solver which loads sparse matrices from a text file in Matrix Market Exchange (.mtx) Format (*Matrix Market: File Formats* 2013) into memory, invokes the solver interfaces, and outputs the results returned from the solver. The tests described below will verify that such a "driver" is functional.

T1 matrix-io

Output: The elements of $\mathbf{A}$ matches exactly the one in the .mtx file. Result solution $\mathbf{x}$ is of size 100.

Result: Pass

3.0.2 Correctness Tests with Manufactured Solutions

This section covers one of the ways to verify the requirements R1 and R2 of the SRS. This includes tests on the accuracy of the solution from the solver by manufacturing an exact solution $\mathbf{x}_{\text{ref}}$ to the problem $\mathbf{Ax} = \mathbf{b}$. This manufacturing process loosely follows the scheme below:

1. $\mathbf{x}_{\text{ref}} \leftarrow$ some random vector
2. $\mathbf{b} \leftarrow \mathbf{A}\mathbf{x}_{\text{ref}}$
3. Solve $\mathbf{Ax} = \mathbf{b}$
4. $e \leftarrow \dfrac{\|\mathbf{x} - \mathbf{x}_{\text{ref}}\|_2}{\|\mathbf{x}_{\text{ref}}\|_2}$

The relative error e will be used as the accuracy metric. The values of the manufactured reference solution $\mathbf{x}_{\text{ref}}$ in this section is uniformly distributed in the range of $[\min(a_{i,j}), \max(a_{i,j})]$. For the test cases in Sections 3.0.2, 3.0.3, and below, the `bundle1` matrix (M. Lourakis 2006) from the Florida

Sparse Matrix Collection (Davis and Hu 2011) will be used as the input matrix $\mathbf{A}$. This matrix has a size of $10\,581 \times 10\,581$ and $770\,811$ non-zeros. The estimated condition number is 1.3×10^4.

T2: generated-double-double

Output: $\mathbf{x}$ with $e = \dfrac{\|\mathbf{x} - \mathbf{x}_{\text{ref}}\|_2}{\|\mathbf{x}_{\text{ref}}\|_2} < 1 \times 10^{-10}$

Result: Pass

T3: generated-single-double

Output: $\mathbf{x}$ with $e = \dfrac{\|\mathbf{x} - \mathbf{x}_{\text{ref}}\|_2}{\|\mathbf{x}_{\text{ref}}\|_2} < 1 \times 10^{-10}$

Result: Pass

3.0.3 Correctness Tests against Trusted Solvers

This section covers the other way to verify the requirements R1 and R2 of the SRS. This includes tests on the accuracy of the yielded solution from the solver by comparing it to an external, trusted solver to the problem $\mathbf{Ax} = \mathbf{b}$. This process loosely follows the scheme below:

1. $\mathbf{x}_{\text{ref}} \leftarrow$ solution by an external solver
2. Solve $\mathbf{Ax} = \mathbf{b}$
3. $e \leftarrow \dfrac{\|\mathbf{x} - \mathbf{x}_{\text{ref}}\|_2}{\|\mathbf{x}_{\text{ref}}\|_2}$

The relative error e will be used as the accuracy metric. For the test cases in this Section, MATLAB® will be used as the external reference solver.

T4: external-double-double

Output: $\mathbf{x}$ with $e = \dfrac{\|\mathbf{x} - \mathbf{x}_{\text{ref}}\|_2}{\|\mathbf{x}_{\text{ref}}\|_2} < 1 \times 10^{-10}$

Result: Pass

T5: external-single-double

Output: $\mathbf{x}$ with $e = \dfrac{\|\mathbf{x} - \mathbf{x}_{\text{ref}}\|_2}{\|\mathbf{x}_{\text{ref}}\|_2} < 1 \times 10^{-10}$

Result: Pass

4 Nonfunctional Requirements Evaluation

4.1 Accuracy

The accuracy of the solver will be assessed by verifying that it converges to a solution within the user-defined tolerance ϵ. The level of accuracy required for computational science and engineering applications will be evaluated through the relative residual norm after convergence. The functional tests T2, T3, T4, T5 are sufficient to verify the nonfunctional requirement NFR1 in the SRS with an accuracy metric of $\epsilon \approx 1 \times 10^{-10}$. Considering that the residual precision $u_r =$ `double`, the chosen $\epsilon \approx 1 \times 10^{-10}$ is reasonably close to the machine epsilon in `double` precise $\epsilon_{\text{mach}} \approx 1.1 \times 10^{-16}$.

4.2 Usability

The usability of the solver will be evaluated based on the clarity and accessibility of its public Application Programming Interface (API). The API should be self-contained, readable, and easy to integrate into other software as a dependency. Usability testing will reference the user characteristics section and include developer feedback. The following tests will be performed to verify the nonfunctional requirement NFR2 in the SRS:

T6: nfr-use

Output/Result: Pass

The usability survey feedback was largely positive, highlighting the API's clarity and ease of integration, with a few suggestions for expanding documentation and use-case coverage. See Appendix A for a detailed report.

4.3 Performance

4.4 etc.

5 Comparison to Existing Implementation

This section will not be appropriate for every project.

6 Unit Testing

7 Changes Due to Testing

8 Automated Testing

9 Trace to Requirements

10 Trace to Modules

11 Code Coverage Metrics

References

Davis, Timothy A. and Yifan Hu (Dec. 7, 2011). "The university of Florida sparse matrix collection". In: *ACM Trans. Math. Softw.* 38.1, 1:1–1:25. ISSN: 0098-3500. DOI: 10.1145/2049662.2049663. URL: https://doi.org/10.1145/2049662.2049663 (visited on 04/11/2025).

M. Lourakis (2006). *Lourakis/bundle1*. SuiteSparse Matrix Collection. URL: https://sparse.tamu.edu/Lourakis/bundle1 (visited on 04/11/2025).

Matrix Market: File Formats (Aug. 14, 2013). The Matrix Market. URL: https://math.nist.gov/MatrixMarket/formats.html (visited on 02/14/2025).

Ye, Xunzhou (2025). *SRS · yex33/MPIR*. GitHub. URL: https://github.com/yex33/MPIR/blob/main/docs/SRS/SRS.pdf.

Appendices

A Usability Survey Results

A.1 Summary

A usability evaluation was conducted on the solver's public API by three VnV team members with experience in numerical software development. Participants reviewed the API documentation, examined usage examples, and completed a structured survey comparing the solver's API to Eigen's sparse solver interface.

The feedback was largely positive, highlighting the API's clarity and ease of integration, with a few suggestions for expanding documentation and use-case coverage.

A.2 Participant Summary

Reviewer	Background
Reviewer A	MSc student, numerical linear algebra
Reviewer B	Senior undergrad, software engineering
Reviewer C	TA, scientific computing

A.3 Likert Scale Results

Question	Very Good	Good	Neutral	Poor
Q1. Intuitive function names	2	1	0	0
Q2. Clear documentation	2	1	0	0
Q3. Sufficient examples	1	2	0	0
Q4. Ease of setup	1	2	0	0
Q5. Quality of error messages	1	2	0	0
Q7. Overall usability	2	1	0	0
Q9. Recommend API	2	1	0	0

A.4 Open-Ended Feedback Highlights

- *Q6. Encountered difficulties:*
  - "It wasn't immediately clear how to switch between CSC and CSR modes."
  - "I expected a higher-level wrapper function that hides the solver config setup."
- *Q8. Suggestions for improvement:*
  - "Add more detailed documentation for return types and default configurations."
  - "Include an end-to-end integration example with a real-world sparse matrix."

A.5 Summary Graphs

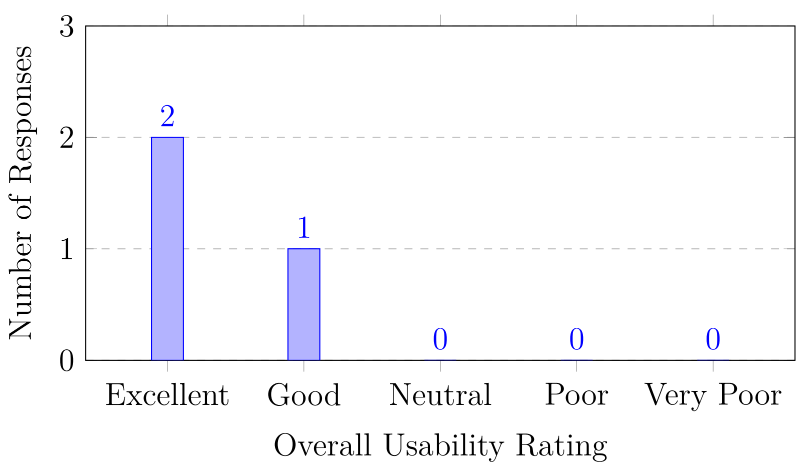

Figure 1: Overall Usability Rating

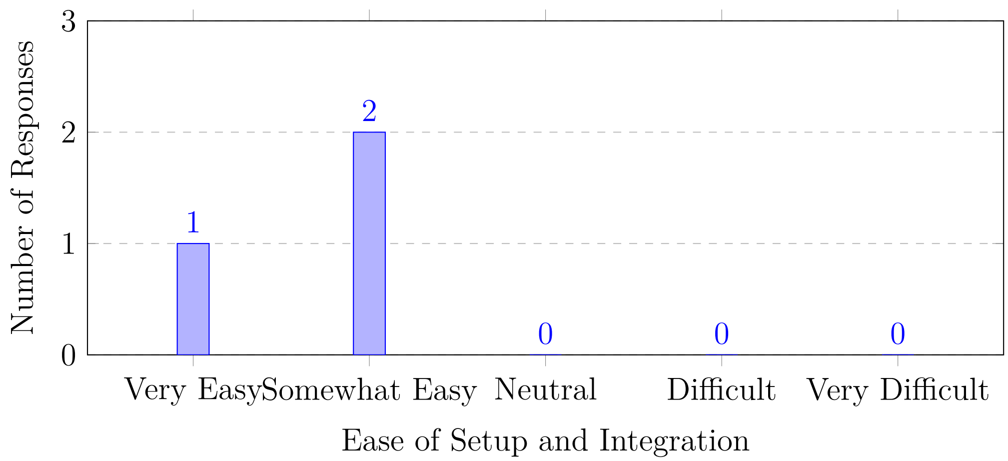

Figure 2: Ease of Setup and Integration

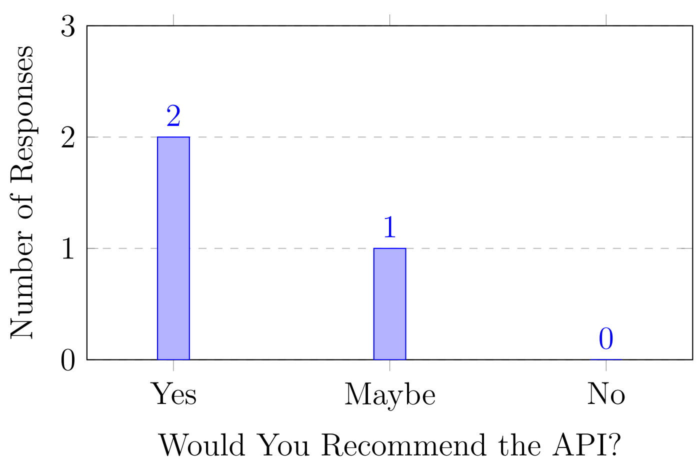

Figure 3: Recommendation Likelihood

A.6 Conclusion

The solver's API was rated as clear and usable, with minor improvements suggested for documentation and configurability. All reviewers would recommend it for use in larger numerical projects, provided additional examples and higher-level wrappers are included.

B Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Reflection.

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response: