Software Requirements Specification for MPIR: A Sparse Linear Solver

Xunzhou (Joe) Ye

February 5, 2025

Contents

Revision History

Date	Version	Notes
26 January 2025	1.0	Initial draft

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

Not applicable to this project because it does not interact with any physical system.

1.2 Table of Symbols

The table that follows summarizes the symbols used in this document. The choice of symbols was made to be consistent with common numerical computing literatures.

symbol	**description**
n	The size of a vector/matrix
$\mathbf{A}$	An $n \times n$ matrix to be solved
$\mathbf{b}$	Some n-vector
ϵ	a solution is found if the norm of the residual is less than ϵ
n_{iter}	the maximum number of iterations to perform
u_f	factorization precision
u_w	working precision
u_r	precision in which the residuals are computed

1.3 Abbreviations and Acronyms

symbol	description
A	Assumption
DD	Data Definition
GD	General Definition
GMRES	General Minimal Residual Method
GS	Goal Statement
IM	Instance Model
IR	Iterative refinement
LC	Likely Change
MP	Mixed-precision
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
TM	Theoretical Model

1.4 Mathematical Notation

2 Introduction

Solving linear systems is a key part of numerical computing and is widely used in many applications, creating a demand for fast and reliable solvers. Modern hardware and software have shown that using lower-precision calculations can be much faster and more memory-efficient than traditional double-precision (64-bit) methods. Mixed-precision (MP) algorithms combine the speed of lower-precision calculations with the accuracy of higher-precision ones to improve performance in various linear solvers. The document describes the solver called MPIR, which implements an iterative method in mixed-precisions and uses GMRES in particular for find the error correction vector during the refinement process.

The following section provides an overview of the Software Requirements Specification (SRS) for MPIR. This section explains the purpose of this document, the scope of the requirements, the characteristics of the intended reader, and the organization of the document.

2.1 Purpose of Document

The primary purpose of this document is to record the requirements of MPIR. Goals, assumptions, theoretical models, definitions, and other model derivation information are specified, allowing the reader to fully understand and verify the purpose and scientific basis of MPIR.

2.2 Scope of Requirements

2.3 Characteristics of Intended Reader

2.4 Organization of Document

3 General System Description

This section provides general information about the system. It identifies the interfaces between the system and its environment, describes the user characteristics and lists the system constraints.

3.1 System Context

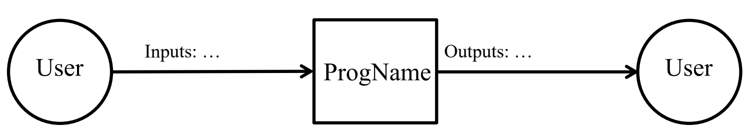

Figure 1: System Context

- User Responsibilities:
  -
- MPIR Responsibilities:
  - Detect data type mismatch, such as a string of characters instead of a floating point number
  -

3.2 User Characteristics

3.3 System Constraints

4 Specific System Description

This section first presents the problem description, which gives a high-level view of the problem to be solved. This is followed by the solution characteristics specification, which presents the assumptions, theories, definitions and finally the instance models.

4.1 Problem Description

MPIR is a solver intended to find a numerical solution $\mathbf{x}$ to a sparse linear system characterized in the equation $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is an $n \times n$ matrix and $\mathbf{b}$ is an n vector.

4.1.1 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

-

4.1.2 Physical System Description

Not applicable to this project because it does not interact with any physical system.

4.1.3 Goal Statements

Given an $n \times n$ matrix $\mathbf{A}$ and an n vector $\mathbf{b}$, where n is the size of the matrix, the goal statement is:

GS1: find a numerical solution $\mathbf{x}$ to the linear system characterized in the equation $\mathbf{Ax} = \mathbf{b}$.

4.2 Solution Characteristics Specification

This section specifies the information in the solution domain of MPIR. This section is intended to express what is required in such a way that analysts and stakeholders get a clear picture, and the latter will accept it. The purpose of this section is to reduce the problem into one expressed in mathematical terms.

This section presents the solution characteristics by successively refining models. It starts with the abstract/general Theoretical Models (TMs) and refines them to the concrete/specific Instance Models (IMs). The instance models that govern MPIRare presented in Subsection 4.2.9. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

4.2.1 Types

4.2.2 Scope Decisions

The scope of this project is inherently limited due to the need to build upon the previous research work rather than formulating a completely new approach to the problem described in Subsection 4.1. A signification portion of this project is dedicated to refactor the prior work, ensuring consistency with its methodology and results before introducing any new contributions. This project must loosely adhere to the same assumptions established in the original study to maintain comparability and validity, as altering these assumptions would fundamentally change the research direction.

4.2.3 Modelling Decisions

4.2.4 Assumptions

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the theoretical model [TM], general definition [GD], data definition [DD], instance model [IM], or likely change [LC], in which the respective assumption is used.

A1:

4.2.5 Theoretical Models

This section focuses on the general equations and laws that MPIR is based on.

RefName: **TM:LDL**

Label: $\mathbf{LDL}^\intercal$ factorization

Equation: $\mathbf{PKP}^\intercal = \mathbf{LDL}^\intercal$

Description: Symmetric quasi-definite matrices are strongly factorizable. For every permutation $\mathbf{P}$ there is a diagonal $\mathbf{D}$ and a unit lower-triangular $\mathbf{L}$ such that the above equation holds.

Notes: Definitions of symmetric matrices and quasi-definite matrices are given in DD1 and DD3, respectively.

Source: Vanderbei 1995, Gill, Saunders, and Shinnerl 1996

Ref. By: GD**??**

Preconditions for TM:LDL: Matrix $\mathbf{A}$ is symmetric quasi-definite.

Derivation for TM:LDL: Not Applicable

RefName: **TM:Krylov**

Label: Krylov subspace

Equation: $\mathbf{K}_m(\mathbf{A}, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \ldots, \mathbf{A}^{m-1}\mathbf{r}_0\}$

Description: $\mathbf{A}$ is any given real, $n \times n$ matrix. In the context of solving $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ is the initial residual. $\mathbf{K}_m(\mathbf{A}, \mathbf{r}_0)$ is said to be "the Krylov subspace of $\mathbf{A}$ of order m with respect to $\mathbf{r}_0$"

Notes: In numerical linear algebra, the Krylov subspace is a sequence of subspaces generated by repeatedly multiplying a matrix $\mathbf{A}$ with an initial vector $\mathbf{r}_0$. It is fundamental to many iterative methods for solving large linear systems and eigenvalue problems.

Source: Ascher and Greif 2011, p. 186

Ref. By: GD**??**

Preconditions for TM:Krylov: None

Derivation for TM:Krylov: Not Applicable

RefName: **TM:Precond**

Label: Matrix Preconditioning

Equation: $\mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b}$ or $\mathbf{A}\mathbf{M}^{-1}\mathbf{y} = \mathbf{b}, \mathbf{x} = \mathbf{M}^{-1}\mathbf{y}$

Description: $\mathbf{A}$ is any given real, $n \times n$ matrix. $\mathbf{M}$ is the preconditioner, chosen such that $\mathbf{M}^{-1}\mathbf{A}$ is better conditioned than $\mathbf{A}$. $\mathbf{M}$ should be easy to invert or solve efficiently.

Notes: Preconditioning is a technique used in iterative methods for solving linear systems to improve convergence speed and numerical stability. It involves transforming the original system into an equivalent one that is easier to solve, typically by reducing the condition number of the matrix.

Source: Ascher and Greif 2011, p. 187

Ref. By: GD**??**

Preconditions for TM:Precond: None

Derivation for TM:Precond: Not Applicable

4.2.6 General Definitions

This section collects the laws and equations that will be used in building the instance models.

Number	GD1
Label	**Iterative refinement in mixed-precision (MP)**
Algorithm	
Description	Newton's law of cooling describes convective cooling from a surface. The law is stated as: the rate of heat loss from a body is proportional to the difference in temperatures between the body and its surroundings. $q(t)$ is the thermal flux ($\mathrm{W\,m^{-2}}$). h is the heat transfer coefficient, assumed independent of T (A**??**) ($\mathrm{W\,m^{-2}\,{}^{\circ}C^{-1}}$). $\Delta T(t) = T(t) - T_{\text{env}}(t)$ is the time-dependent thermal gradient between the environment and the object (°C).
Source	Citation here
Ref. By	DD**??**, DD**??**

Detailed derivation of simplified rate of change of temperature

4.2.7 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given.

Number	DD1
Label	**Symmetric matrix**
Equation	$\mathbf{A}^{\intercal} = \mathbf{A}$
Description	A square matrix is symmetric if it satisfies the equation above.
Sources	Ascher and Greif 2011, p. 78
Ref. By	DD3

Number	DD2
Label	**Positive definite matrix**
Equation	$\mathbf{x}^\intercal\mathbf{A}\mathbf{x} > 0 \quad \forall\mathbf{x} \neq 0$
Description	A square matrix is positive definite if it satisfies the equation above. For any column vector $\mathbf{x} = (x_1, \ldots, x_n)^\intercal$, we require $\sum_{i,j=1}^{n} a_{i,j}x_ix_j > 0$, provided that at least one component $x_j \neq 0$.
Sources	Ascher and Greif 2011, p. 78
Ref. By	DD3

Number	DD3
Label	**Symmetric quasi-definite matrix**
Equation	$\mathbf{K} = \begin{bmatrix} -\mathbf{E} & \mathbf{A}^\intercal \\ \mathbf{A} & \mathbf{F} \end{bmatrix}$
Description	A symmetric matrix (defined in DD1) $\mathbf{K}$ is quasi-definite if it has the above form. $\mathbf{E}, \mathbf{F}$ are symmetric positive definite matrices (defined in DD2).
Sources	Gill, Saunders, and Shinnerl 1996
Ref. By	TM4.2.5

4.2.8 Data Types

This section collects and defines all the data types needed to document the models.

Type Name	Name for Type
Type Def	mathematical definition of the type
Description	description here
Sources	Citation here, if the type is borrowed from another source

4.2.9 Instance Models

This section transforms the problem defined in Section 4.1 into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 4.2.7 to replace the abstract symbols in the models identified in Sections 4.2.5 and 4.2.6.

The goals are solved by .

Number	IM1
Label	**Energy balance on water to find T_W**
Input	m_W, C_W, h_C, A_C, h_P, A_P, t_{final}, T_C, T_{init}, $T_P(t)$ from IM**??** The input is constrained so that $T_{\text{init}} \leq T_C$ (A**??**)
Output	$T_W(t)$, $0 \leq t \leq t_{\text{final}}$, such that $\frac{dT_W}{dt} = \frac{1}{\tau_W}[(T_C - T_W(t)) + \eta(T_P(t) - T_W(t))]$, $T_W(0) = T_P(0) = T_{\text{init}}$ (A**??**) and $T_P(t)$ from IM**??**
Description	T_W is the water temperature (°C). T_P is the PCM temperature (°C). T_C is the coil temperature (°C). $\tau_W = \frac{m_W C_W}{h_C A_C}$ is a constant (s). $\eta = \frac{h_P A_P}{h_C A_C}$ is a constant (dimensionless). The above equation applies as long as the water is in liquid form, $0 < T_W < 100^o$C, where 0^oC and 100^oC are the melting and boiling points of water, respectively (A**??**, A**??**).
Sources	Citation here
Ref. By	IM**??**

Derivation of ...

4.2.10 Input Data Constraints

Table 1 shows the data constraints on the input output variables. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. The column for software constraints restricts the range of inputs to reasonable values. The software constraints will be helpful in the design stage for picking suitable algorithms. The constraints are conservative, to give the user of the model the flexibility to experiment with unusual situations. The column of typical values is intended to provide a feel for a common scenario. The uncertainty column provides an estimate of the confidence with which the physical quantities can be measured. This information would be part of the input if one were performing an uncertainty quantification exercise.

The specification parameters in Table 1 are listed in Table 2.

(*)

Table 1: Input Variables

Var	Physical Constraints	Software Constraints	Typical Value	Uncertainty
L	$L > 0$	$L_{\text{min}} \leq L \leq L_{\text{max}}$	1.5 m	10%

Table 2: Specification Parameter Values

Var	Value
L_{min}	0.1 m

4.2.11 Properties of a Correct Solution

A correct solution must exhibit .

Table 3: Output Variables

Var	Physical Constraints
T_W	$T_{\text{init}} \leq T_W \leq T_C$ (by A**??**)

5 Requirements

This section provides the functional requirements, the business tasks that the software is expected to complete, and the nonfunctional requirements, the qualities that the software is expected to exhibit.

5.1 Functional Requirements

R1:

R2:

R3:

R4:

R5:

5.2 Nonfunctional Requirements

NFR1: **Accuracy**

NFR2: **Usability**

NFR3: **Maintainability**

NFR4: **Portability**

- Other NFRs that might be discussed include verifiability, understandability and reusability.

5.3 Rationale

6 Likely Changes

LC1:

7 Unlikely Changes

LC2:

8 Traceability Matrices and Graphs

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the column of that component that are marked with an "X" may have to be modified as well. Table 4 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 5 shows the dependencies of instance models, requirements, and data constraints on each other. Table 6 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions.

The purpose of the traceability graphs is also to provide easy references on what has to be additionally modified if a certain component is changed. The arrows in the graphs represent dependencies. The component at the tail of an arrow is depended on by the component at the head of that arrow. Therefore, if a component is changed, the components that it points to should also be changed. Figure **??** shows the dependencies of theoretical models, general definitions, data definitions, instance models, likely changes, and assumptions on each other.

	TM??	TM??	TM??	GD1	GD??	DD??	DD??	DD??	DD??	IM1	IM??	IM??
TM??												
TM??			X									
TM??												
GD1												
GD??	X											
DD??				X								
DD??				X								
DD??												
DD??								X				
IM1					X	X	X				X	
IM??					X		X		X	X		
IM??		X										
IM??		X	X				X	X	X		X	

Table 4: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM1	IM??	IM??	IM??	4.2.10	R??	R??
IM1		X				X	X
IM??	X			X		X	X
IM??						X	X
IM??		X				X	X
R??							
R??						X	
R??					X		
R2	X	X				X	X
R??	X						
R??		X					
R??			X				
R??				X			
R4			X	X			
R??		X					
R??		X					

Table 5: Traceability Matrix Showing the Connections Between Requirements and Instance Models

	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??
TM??	X																		
TM??																			
TM??																			
GD1		X																	
GD??			X	X	X	X													
DD??							X	X	X										
DD??			X	X						X									
DD??																			
DD??																			
IM1											X	X		X	X	X			X
IM??												X	X			X	X	X	
IM??														X					X
IM??													X					X	
LC??				X															
LC??								X											
LC??									X										
LC??											X								
LC??												X							
LC??															X				

Table 6: Traceability Matrix Showing the Connections Between Assumptions and Other Items

Figure **??** shows the dependencies of instance models, requirements, and data constraints on each other.

9 Development Plan

10 Values of Auxiliary Constants

References

Ascher, Uri M and Chen Greif (Jan. 2011). *A first course on numerical methods*. 3600 Market Street, 6th Floor Philadelphia, PA 19104-2688: SIAM.

Gill, Philip E., Michael A. Saunders, and Joseph R. Shinnerl (Jan. 1996). "On the Stability of Cholesky Factorization for Symmetric Quasidefinite Systems". In: *SIAM Journal on Matrix Analysis and Applications* 17.1, pp. 35–46. ISSN: 0895-4798, 1095-7162. DOI: 10.1137/S0895479893252623. URL: http://epubs.siam.org/doi/10.1137/S0895479893252623 (visited on 02/05/2025).

Vanderbei, Robert J. (Feb. 1995). "Symmetric Quasidefinite Matrices". In: *SIAM Journal on Optimization* 5.1, pp. 100–113. ISSN: 1052-6234, 1095-7189. DOI: 10.1137/0805005. URL: http://epubs.siam.org/doi/10.1137/0805005 (visited on 02/05/2025).