Software Requirements Specification for ChemCode: A program for solving chemistry problems

Samuel Crawford

March 6, 2023

Contents

1 Revision History

Date	Version	Notes
Jan. 21, 2023	0.0	Start document, fill in title page, and add references
	0.1.0	Fill in Problem Description and Goal Statements sections, as well as all relevant definitions
Jan. 22, 2023	0.1.1	Replace the notion of "fractional oxidation state" with "non-stoichiometric compound"
	0.1.2	Fix capTemplate reference
	0.2.0	Update Abbreviations and Acronyms
	0.2.1	Add IM for balancing a chemical reaction, along with relevant reference information
Jan. 23, 2023	0.2.2	Add IM for determining if a chemical reaction is feasible, along with relevant reference information
Jan. 25, 2023	0.2.3	Add references between sections for notation and definitions
	0.2.4	Abstract IM for determining if a chemical reaction is feasible
	0.2.5	Add IM for converting chemical equation to a matrix, along with relevant reference information
	0.2.6	Rename and add input constraint to IM for balancing a chemical reaction
	0.2.7	Add note about IM for balancing a chemical reaction returning the smallest solution
	0.2.8	Add TM for matrix equation (TM1)
	0.2.9	Improve referencing, mainly for TMs, and add note about potential TM for Conservation of Mass
	0.3.0	Add check for Conservation of Mass in IM for determining if a chemical reaction is feasible
Jan. 26, 2023	0.3.1	Add Likely Changes from Problem Statement, add assumption about difference between number of elements and compounds (A1), and add mole to Table of Units
	0.3.2	Update TM1 (and TM function) from feedback from presentation
	0.3.3	Fill in Scope of Requirements, fix A1 and change "infeasible" to "feasible"
Jan. 27, 2023	0.4.0	Improve section referencing in Revision History
	0.4.1	Fill in Functional Requirements
Jan. 28, 2023	0.4.2	Fill in main Nonfunctional Requirements, along with relevant reference information
	0.4.3	Add TM for the Law of Conservation of Mass (TM2)

Date	Version	Notes
Jan. 28, 2023	0.4.4	Represent input to IM for converting chemical equation to a matrix as two sequences of chemical formulas, add assumptions about user input, and remove infeasible likely change
Jan. 29, 2023	0.5.0	Add NFRs for understandability (NFR2) and verifiability (NFR6) and improve quality of other NFRs
	0.5.1	Fill in Introduction based on Drasil examples
	0.5.2	Fill in Physical System Description and remove "product" and "reactant" from Terminology and Definitions
Jan. 30, 2023	0.5.3	Remove unnecessary template comments about SRS document and the ones from Specific System Description up to Instance Models
	0.5.4	Remove unnecessary content from Table of Symbols and add the notion of "gram" to the document
	0.6.0	Make NFR1 more accurate to current scope with potential future information for stretch goals added in comment
	0.6.1	Make System Context diagram (Figure 1) and start to fill in System Context
Jan. 31, 2023	0.6.2	Fill in rest of General System Description
Feb. 1, 2023	0.6.3	Change abbreviation of theoretical model from "T" to "TM" and make TM references consistent with other references
Feb. 2, 2023	0.6.4	Fill in Reference Material, renaming specific matrices/vectors in Instance Models
	0.6.5	Reformat and clarify Likely Changes
Feb. 3, 2023	0.6.6	Add type notation and information
	0.6.7	Fill in Unlikely Changes, add likely change for identifying more types of reactions (LC9), and reference A1 in IM for converting chemical equation to a matrix
	0.6.8	Fill in Traceability Matrices [To be updated —SC] and remove more unnecessary content
	1.0	Fill in data constraints sections, update traceability matrices, and finalize formatting
Feb. 14, 2023	1.1.0	Improve mathematical notation (remove "$\times 1$" from vector types, define $\mathbb{R}$) and citations
	1.2.0	Improve requirements based on feedback
Feb. 22, 2023	1.2.1	Add equation to TM2, including relevant Data Types
	1.2.2	Improve TM1 based on feedback
	1.2.3	Add ChemCode abbreviation and missing definitions
Feb. 26, 2023	1.2.4	Add DD1
Feb. 27, 2023	1.2.5	Add DD3 and clarify Mathematical Notation for types
Feb. 28, 2023	1.2.6	Add DD2
	1.2.7	Add GD1

Date	Version	Notes
Feb. 28, 2023	1.2.8	Add DD for GCF (DD4)
	1.2.9	Add notion of parametric solution to GD1 and add IM1
	1.3.0	Remove old IMs
Mar. 4, 2023	1.3.1	Clarify inputs and outputs and improve line breaking in System Context
	1.3.2	Replace method of solving with integer programming and improve Revision History table format
Mar. 5, 2023	1.3.3	Replace old TM1 with one for an integer linear program (TM1)
	1.3.4	Replace old GD1 with one that corresponds to TM1 (GD1)
Mar. 6, 2023	1.3.5	Clarify GD1 (types, output, and special matrices/vectors)
	1.3.6	Remove all content from specification based on solving a system of linear equations

2 Reference Material

This section records information for easy reference.

2.1 Abbreviations and Acronyms

symbol	description
A	Assumption
ChemCode	Chemistry Code
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
TM	Theoretical Model
VnV	Verification and Validation

2.2 Table of Units

Throughout this document, SI (Système International d'Unités) is employed as the unit system. Several units are used as described below. For each unit, the symbol is given, followed by a description of the unit and the SI name.

symbol	unit	SI
g	mass	gram
mol	amount of substance	mole

2.3 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units and types. The choice of symbols was made to be consistent with linear algebra conventions while distinguishing between specific matrices. The symbols are listed in alphabetical order.

[Should symbols for data types be included here? —SC]

symbol	unit	type	description
$\mathbf{0}$	–	$\mathbb{Z}_{\geq 0}^{m}$	zero vector
$\mathbf{1}$	–	$\mathbb{Z}_{>0}^{m}$	unary vector
$\mathbf{A}$	–	$\mathbb{R}^{m \times n}$	generic matrix
$\mathbf{b}$	–	$\mathbb{R}^{m}$	generic vector
$\mathbf{c}$	–	$\mathbb{Z}_{>0}^{n}$	vector of the smallest coefficients that balance a chemical equation
c	–	C	generic compound
$\mathbf{E}$	–	$\mathbb{R}^{m \times n}$	matrix representation of a chemical equation
e	–	E	generic element
i	–	$\mathbb{Z}_{\geq 0}$	generic non-negative integer
j	–	$\mathbb{Z}_{\geq 0}$	generic non-negative integer
m	–	$\mathbb{Z}_{>0}$	generic positive integer
n	–	$\mathbb{Z}_{>0}$	generic positive integer
P	–	$\langle C \rangle$	product side of a chemical equation
R	–	$\langle C \rangle$	reactant side of a chemical equation
$\mathbf{x}$	–	$\mathbb{R}^{n}$	generic vector
x	–	$\mathbb{Z}_{\geq 0}$	generic non-negative integer
z	–	$\{\mathbb{Z}\}$	generic set of integers

2.4 Mathematical Notation

The set of non-negative integers (including zero) is denoted by $\mathbb{Z}_{\geq 0}$ and the set of positive integers (excluding zero) is denoted by $\mathbb{Z}_{>0}$ [1]. The set of real numbers is denoted by $\mathbb{R}$ and consists of every number "that can be expressed as an infinite decimal expansion" [2]. The sets of non-negative (including zero) and positive (excluding zero) real numbers are denoted by $\mathbb{R}_{\geq 0}$ and $\mathbb{R}_{>0}$, respectively [3].

Matrices are written in bold uppercase letters (for example $\mathbf{A}$), while vectors (one-dimensional matrices) are written in bold lowercase letters, (for example $\mathbf{x}$) [4]. The type of a matrix will be denoted by the type of each value in the matrix with the matrix's dimensions as a superscript; for example, a matrix of real numbers with two rows and three columns is of type $\mathbb{R}^{2 \times 3}$ [5, p. 3]. Since vectors are one-dimensional matrices, their types only have one dimension; for example, a row vector with four real number entries is denoted by $\mathbb{R}^{4}$ [5,

p. 4]. The zero matrix/vector is a special type of matrix/vector where each element is zero and is denoted by $\mathbf{0}$ [6]. Similarly, the unary matrix/vector, denoted by $\mathbf{1}$, will represent a matrix/vector where each element is one. Individual entries in a matrix are written in lowercase letters with their row and column number, in that order, written as subscripts (for example a_{12}) [4]. Two (or more) matrices with the same number of rows can be joined together to form an "augmented matrix", written as $(\mathbf{A}|\mathbf{B})$ [7]. [Add notation for transpose and identity matrix —SC]

The notation for definite description ($\mathbf{I}$) is taken from [8].

3 Introduction

Chemical equations are common ways of representing chemical reactions (as described in Physical System Description), but must be balanced before they can be used [9]. Therefore, it is useful to have a tool to automatically balance these chemical reactions to improve the productivity of scientists and engineers and reduce the potential for human error.

The following section provides an overview of the Software Requirements Specification (SRS) for ChemCode. 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.

3.1 Purpose of Document

The primary purpose of this document is to record the requirements of the program for solving chemistry problems [Is this a good description? —SC]. 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 ChemCode. With the exception of system constraints, this SRS will remain abstract, describing what problem is being solved, but not how to solve it.

This document will be used as a starting point for subsequent development phases, including writing the design specification and the software Verification and Validation (VnV) Plan. The design document will show how the requirements are to be realized, including decisions on the numerical algorithms and programming environment. The VnV Plan will show the steps that will be used to increase confidence in the software documentation and the implementation. Although the SRS fits in a series of documents that follow the so-called "waterfall model," the actual development process is not constrained in any way. Even when the waterfall model is not followed, the most logical way to present the documentation is to "fake" a rational design process [10].

3.2 Scope of Requirements

The scope of the requirements includes determining if a given chemical equation is feasible (see Terminology and Definitions) and, if it is, balancing it using the smallest positive integer coefficients possible, regardless of whether or not the chemical formulas in the equation are correct or if the reaction would take place in a real-world setting.

3.3 Characteristics of Intended Reader

Reviewers of this documentation should have an understanding of high-school level chemistry, namely stoichiometry (see Terminology and Definitions), and undergraduate Level 3 linear optimization, namely integer programming.

3.4 Organization of Document

The organization of this document is based on a template from [11], which is based on the template for an SRS for scientific computing software proposed by [12], [13], and [14]. The presentation follows the standard pattern of presenting goals, theories, definitions, and assumptions. For readers that would like a more bottom-up approach, they can start reading the Instance Models and trace back to find any additional information they require.

The Goal Statements are refined to the Theoretical Models, which are refined to the Instance Models.

4 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.

4.1 System Context

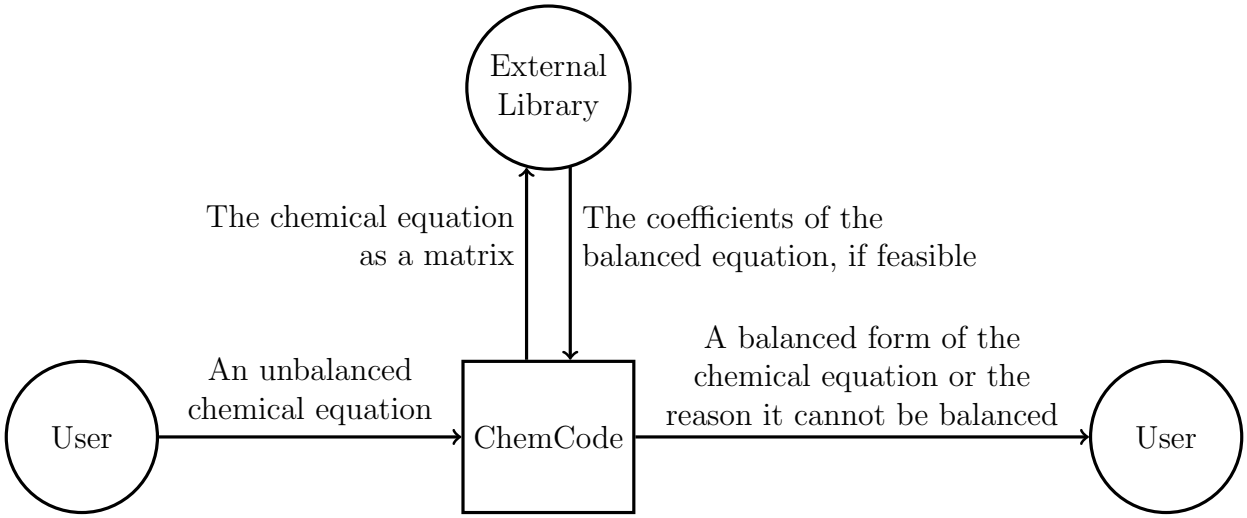

Figure 1: System Context

Figure 1 shows the system context. A circle represents an external entity outside the software. A rectangle represents the software system itself (ChemCode). Arrows are used to show the data flow between the system and its environment.

- User Responsibilities:
  - Provide an unbalanced chemical equation, ensuring conformation to input data format required by ChemCode

  - Ensure required software assumptions from Subsection 5.2.1 are appropriate for the problem to which the user is applying the software

- ChemCode Responsibilities:
  - Detect data type mismatch, such as a string of characters instead of a rational number
  - Detect improperly formatted chemical equations
  - Format the inputted chemical equation as a matrix
  - If the inputted chemical equation is feasible, find its balanced form with the smallest positive integer coefficients
  - If the inputted chemical equation is infeasible, determine the reason why

- External Library Responsibilities:
  - Solve the integer programming problem for the chemical equation

4.2 User Characteristics

The end user of ChemCode should have an understanding of high-school level chemistry, namely stoichiometry (see Terminology and Definitions).

4.3 System Constraints

ChemCode will be developed using Drasil [15], "a framework for generating high-quality documentation and code for Scientific Computing Software" [16, p. iii], with the goal of extending it by adding concepts relevant to the problem outlined in Problem Description. Since Drasil is built on the idea of reusability, external libraries will be used to solve the integer programming problems. This was previously done with ordinary differential equation (ODE) solvers, since "creating a complete ODE solver in Drasil would take considerable time, and there are already many reliable external libraries . . . tested by long use" [17, p. 24]; these rationales also apply to solvers of integer programming problems.

5 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.

5.1 Problem Description

ChemCode is intended to balance chemical equations (including ones with nonstoichiometric compounds; see Terminology and Definitions) so they can be useful for other computations [9]. Additionally, since molecules only exist in positive integers (since dividing a molecule changes its composition into new types of molecules), the coefficients used to balance the equation must be whole numbers, and by convention should be as small as possible [9].

5.1.1 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:

- **Balanced**: (Referring to a chemical equation) following the Law of Conservation of Matter (TM2) [9].
- **Compound**: A molecule made up of more than one atom, which may or may not be of different elements.
- **Element**: The group of "all atoms with the same number of protons in the atomic nucleus" [18]. For example, all atoms with one proton are hydrogen atoms.
- **Feasible**: (Referring to a chemical equation) able to be balanced [19].
- **Hydrate**: "A compound formed by the chemical combination of water and some other substance in a definite molecular ratio" [20].
- **Nonstoichiometric Compound**: "Any solid chemical compound in which the numbers of atoms of the elements present cannot be expressed as a ratio of small positive integers" [21].
- **Stoichiometry**: "The calculation of the quantities of reactants or products in a chemical reaction using the relationships found in a balanced chemical equation" [9, p. 337].

5.1.2 Physical System Description

Chemical reactions are interactions between different types of matter that result in new substances being formed [9, p. 286]. These are represented using chemical equations, with the reactant(s)—the substance(s) present at the beginning of the reaction—on the left-hand side and the product(s)—the substance(s) formed by the reaction—on the right. An example of a chemical equation representing the combustion of methane in the presence of oxygen to form carbon dioxide and water vapour is shown below. The subscripts indicate how many atoms of each element are present in the given chemical compound.

$$CH_4 + O_2 \rightarrow CO_2 + H_2O$$

As mentioned in Problem Description, chemical formulas must be balanced to be useful [9]. This means that there must be the same number of each element before and after the reaction takes place to satisfy the Law of Conservation of Mass (TM2). Since changing the subscripts in a formula will change the type of molecule (for example, O_3 is ozone, not oxygen), equations are balanced by introducing coefficients before each compound. If no coefficient is present, there is an implicit coefficient of "1". The above equation looks like this in its balanced form:

$$CH_4 + 2\,O_2 \rightarrow CO_2 + 2\,H_2O$$

The physical system of ChemCode includes:

PS1: the products of a given chemical reaction;

PS2: the reactants of the same chemical reaction.

5.1.3 Goal Statements

Given a representation of a chemical equation, the goal statements of ChemCode are:

GS1: Determine if a given chemical reaction is feasible (see Terminology and Definitions).

GS2: Balance the chemical equation with the smallest positive integer coefficients possible.

5.2 Solution Characteristics Specification

The instance models that govern ChemCode are presented in Subsection 5.2.6. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

5.2.1 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], likely change [LC], or unlikely change [UC] in which the respective assumption is used.

A1: For all inputted chemical equations, there is at most one more compound than element. [Should this be an assumption or a requirement? —SC] [GD3, UC1]

A2: All inputted chemical formulas describe real chemical formulas. [GD3]

A3: All inputted chemical formulas are formatted following some set of agreed-upon conventions. [GD3, LC6]

A4: All inputted chemical formulas are simple (i.e., only consist of atomic symbols and subscripts). [GD3, LC1]

5.2.2 Theoretical Models

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

Number: TM1

Label: Integer linear program

Equation:

$$\begin{aligned} &\text{maximize} && \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ &\text{subject to} && \mathbf{A}\mathbf{x} \leq \mathbf{b}, \\ & && \mathbf{x} \geq \mathbf{0}, \\ &\text{and} && \mathbf{x} \in \mathbb{Z}^n \end{aligned}$$

[Is it right for this to be labelled "equation"? —SC]

Description: The above equation [? —SC] gives the canonical form of an integer linear program, which is "a mathematical optimization or feasibility program in which some or all of the variables are restricted to be integers [and] the objective function and the constraints (other than the integer constraints) are linear" [22]. $\mathbf{A}$ is a generic matrix of type $\mathbb{R}^{m \times n}$, $\mathbf{b}$ is a generic vector of type $\mathbb{R}^m$, $\mathbf{c}$ is a generic vector of type $\mathbb{R}^n$, and $\mathbf{x}$ is a generic vector of type $\mathbb{Z}^n$. The values of $\mathbf{x}$ are unknown and will be solved for.

Notes: None

Source: https://en.wikipedia.org/wiki/Integer_programming

Ref. By: GD1

Derivation for TM1: Not Applicable

Number: TM2

Label: Law of Conservation of Mass

Equation: $\forall\, e : E,\ R, P : \langle C \rangle \mid \sum_{c \in R} \text{count}(e, c) = \sum_{c \in P} \text{count}(e, c)$, where R represents the reactant side of a chemical equation, P represents the product side, and $\text{count}(e, c)$ is from DD1.

Description: This law states that "matter can neither be created nor destroyed in a chemical reaction . . . but it may change forms to other substances" [9, p. 112].

Notes: None

Source: https://chem.libretexts.org/Courses/Anoka-Ramsey_Community_College/Introduction_to_Chemistry/03%3A_Matter_and_Energy/3.06%3A_Conservation_of_Mass

Ref. By: GD1

Derivation for TM2: Not Applicable

5.2.3 General Definitions

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

Number	GD1
Label	**Integer linear program for chemical equation**
SI Units	–
Equation	minimize $\mathbf{x}$ subject to $\mathbf{Ex} = \mathbf{0}$, $\mathbf{x} > \mathbf{0}$, and $\mathbf{x} \in \mathbb{Z}^n$ [Is it right for this to be labelled "equation"? —SC]
Description	This is a specific instance of the ILP from TM1 with $\mathbf{A} = \mathbf{E}$ (from DD3), $\mathbf{b} = \mathbf{0}$ (since there should be no difference between the number of atoms of each element on each side of the equation), and $\mathbf{c} = \mathbf{1}$ (since only the sum of the coefficients need to be minimized to obtain the smallest solution). The goal is changed from "maximize" to "minimize" since chemical equations have the smallest coefficients by convention [9]. The constraints are also modified; for the Law of Conservation of Mass (TM2) to hold, $\mathbf{Ex}$ must equal $\mathbf{0}$, and to avoid trivial solutions and ensure that mass is conserved, each coefficient must be positive ($\mathbf{x} > \mathbf{0}$). The solution to this equation [? —SC] is the vector of the smallest positive integer coefficients for the compounds in the chemical equation such that the Law of Conservation of Mass (TM2) is satisfied. $\mathbf{0}$ is the zero vector and $\mathbf{1}$ is the unary vector (see Mathematical Notation).
Source	[19]
Ref. By	IM??

5.2.4 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	**Count of an element in a compound**
Symbol	$\text{count}(e, c)$
SI Units	–
Equation	$\text{count}(e, c) = \begin{cases} \mathbf{I}\, x : \mathbb{R}_{>0} \mid (e, x) \in c, & \exists\, x : \mathbb{R}_{>0} \mid (e, x) \in c \\ 0, & \text{otherwise} \end{cases}$
Description	The output of $\text{count}(e, c)$ is of type $\mathbb{R}_{\geq 0}$ and represents the number of atoms of a given element e (of type E) in a given compound c (of type C).
Sources	–
Ref. By	TM2, DD3

Number	DD2
Label	**Set of elements in a chemical equation**
Symbol	$\text{elems}(R, P)$
SI Units	–
Equation	$\text{elems}(R, P) = \{e : E \mid \exists\, x : \mathbb{R}_{>0}, c \in R \cup P \mid (e, x) \in c\}$
Description	The output of $\text{elems}(R, P)$ is of type $\langle E \rangle$ and represents the set of elements that occur in a given chemical equation where R (of type $\langle C \rangle$) represents its reactant side and P (of type $\langle C \rangle$) represents its product side.
Sources	–
Ref. By	DD3

Number	DD3
Label	**Matrix representation of a chemical equation**
Symbol	**E**
SI Units	–
Equation	$\forall\, e_i \in \text{elems}(R, P), c_j \in R \cup P \mid e_{ij} = \begin{cases} \text{count}(e_i, c_j), & c_j \in R \\ -\text{count}(e_i, c_j), & \text{otherwise} \end{cases}$
Description	**E** is a matrix of type $\mathbb{R}^{\lvert\text{elems}(R,P)\rvert \times \lvert R \cup P\rvert}$ constructed from R, a sequence of compounds ($\langle C \rangle$) on the reactant side of a chemical equation, and P a sequence of compounds ($\langle C \rangle$) on the product side. It represents a given chemical equation as a matrix where each row corresponds to a different element and each column corresponds to a different compound. Positive entries indicate the compound is a reactant, negative entries indicate the compound is a product, and zeroes indicate the corresponding element is not present in the corresponding compound. $\text{count}(e, c)$ is from DD1 and $\text{elems}(R, P)$ is from DD2. Each compound present in the inputted sequences are assumed to be valid (A2), be formatted correctly (A3), and only consist of atomic symbols and subscripts (A4). The number of columns is assumed to be at most one more than the number of rows (A1).
Sources	[19]
Ref. By	GD1, IM??

5.2.5 Data Types

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

Type Name	Element
Type Def	$E = \{$H, He, Li, Be, B, C, N, O, F, Ne, Na, Mg, Al, Si, P, S, Cl, Ar, K, Ca, Sc, Ti, V, Cr, Mn, Fe, Co, Ni, Cu, Zn, Ga, Ge, As, Se, Br, Kr, Rb, Sr, Y, Zr, Nb, Mo, Tc, Ru, Rh, Pd, Ag, Cd, In, Sn, Sb, Te, I, Xe, Cs, Ba, La, Ce, Pr, Nd, Pm, Sm, Eu, Gd, Tb, Dy, Ho, Er, Tm, Yb, Lu, Hf, Ta, W, Re, Os, Ir, Pt, Au, Hg, Tl, Pb, Bi, Po, At, Rn, Fr, Ra, Ac, Th, Pa, U, Np, Pu, Am, Cm, Bk, Cf, Es, Fm, Md, No, Lr, Rf, Db, Sg, Bh, Hs, Mt, Ds, Rg, Cn, Nh, Fl, Mc, Lv, Ts, Og$\}$
Description	A type representing each element from [23].
Sources	[24]

Type Name	Compound
Type Def	$C = \langle (E, \mathbb{R}_{>0}) \rangle$
Description	A type representing a compound (a sequence of elements, each with a specified positive quantity).
Sources	–

5.2.6 Instance Models

This section transforms the problem defined in Problem Description into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 5.2.4 to replace the abstract symbols in the models identified in Sections 5.2.2 and 5.2.3 [Should the reference to the general models be removed if there aren't any? —SC].

The goals GS1 and GS2 are solved by [Add reference to output IM —SC] and IM**??**, respectively.

5.2.7 Input Data Constraints

There are no input data constraints.

5.2.8 Properties of a Correct Solution

The vector outputted by GD1 can be used as the value for $\mathbf{x}$ in a "sanity" check; the original input (formatted as a matrix according to DD3) can then be used as $\mathbf{E}$. Performing this matrix multiplication should result in the zero vector, $\mathbf{0}$ (from GD1).

6 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.

6.1 Functional Requirements

R1: Input a representation of a chemical equation.

R2: Convert the inputted equation to matrix form (from DD3).

R3: Determine if the inputted equation is feasible (see Terminology and Definitions; from [Add ref to output IM —SC]).

R4: If the inputted equation is infeasible, output a descriptive message.

R5: If the inputted equation is feasible, balance the chemical equation with the smallest positive integer coefficients possible (from GD1).

R6: If the inputted equation is feasible, output a balanced form of the equation in the same format as the input (from [Add ref out output IM —SC]).

6.2 Nonfunctional Requirements

NFR1: **Accuracy:** Chemical equations are only useful if they are balanced [9], so computed coefficients from GD1 should be exact.

NFR2: **Understandability:** A new intended user (as described by User Characteristics) should be able to learn how to use ChemCode in an acceptable amount of time, as measured by the procedure in Section X of the Verification and Validation (VnV) Plan.

NFR3: **Usability:** An intended user (as described by User Characteristics) should find ChemCode easy to use, as measured by the procedure in Section X of the VnV Plan.

NFR4: **Maintainability:** The development time for any of the likely changes should not exceed MAINTAIN_FRAC of the original development time.

NFR5: **Portability:** ChemCode should be able to run on systems with the corresponding programming language installed, including systems running on Windows or macOS. The tests from the VnV Plan should pass in these environments.

NFR6: **Verifiability:** ChemCode is tested following the VnV Plan.

7 Likely Changes

LC1: The system currently assumes that inputted chemical formulas are simple (A4). In the future, the user might be able to input more complex chemical formulas, such as hydrates (see Terminology and Definitions) or those with polymers or isotopes.

LC2: In the future, ChemCode might be able to, given the amount of one substance (in moles) in a reaction, calculate the amount required/produced of every other substance (also in moles) in the reaction.[1]

LC3: In the future, ChemCode might be able to, given the amount of each reactant (in moles) in a reaction, determine the limiting reactant(s).[1] This is dependent on LC2.

LC4: In the future, ChemCode might be able to, given the amount of more than one reactant (in moles) in a reaction, calculate the theoretical yield of each product (also in moles).[1] This is dependent on LC3.

LC5: In the future, ChemCode might be able to, given the amount of more than one reactant (in moles) in a reaction, calculate the amount of excess reactant(s) (also in moles).[1] This is dependent on LC3.

LC6: The system currently assumes that inputted chemical formulas are formatted according to some set of conventions (A3). In the future, ChemCode might be able to parse valid but incorrectly formatted chemical formulas inputted by the user and format them correctly when outputting them.

LC7: In the future, the user might be able to enter the amounts required by LC2, LC3, LC4, and LC5 in terms of mass (e.g., in grams).[1]

LC8: In the future, ChemCode might be able to classify a chemical reaction as "combination (or synthesis), decomposition, combustion, single replacement, [or] double replacement" [9, p. 301].[1]

LC9: In the future, ChemCode might also be able to classify "oxidation-reduction reactions, ... acid-base reactions, and condensation reactions" [9, p. 301].[1] This should be done after LC8.

LC10: In the future, ChemCode might allow the user to input phase labels.[1]

LC11: In the future, ChemCode might be able to, given the phase labels for the reactants of a reaction, identify the phase labels for the products, which would involve determining solubility.[1] This is dependent on LC10.

LC12: In the future, ChemCode might be able to identify when a reaction will not take place.[1] This is dependent on LC11.

[1] These examples of problems related to chemical equations were taken from [9].

8 Unlikely Changes

UC1: A1 assumes that for all inputted chemical equations, there is at most one more compound than element. Using WebQC's chemical equation balancer [25], equations without this constraint, such as $O_3 + C \rightarrow O_2 + CO_2$, "can be balanced in an infinite number of ways" and are "combination[s] of two different reactions" [25]. Manual empirical analysis of these equations led to unexpected results. The constraint in A1 might be a law of nature and a byproduct of "balancing chemical equations [being] just linear algebra" [26, p. 193], but preliminary investigation did not find any proof of this. Therefore, this constraint will remain an assumption and is unlikely to be relaxed.

9 Traceability Matrices [To be updated —SC]

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 1 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and data constraints on each other. Table 3 shows the dependencies of theoretical models, general definitions, data definitions, instance models, likely changes, and unlikely changes on the assumptions. Table 2 shows the dependencies of requirements on instance models.

	TM??	TM2	GD3	DD1	IM??
TM??					
TM2				X	
DD1					
GD3					X
IM??	X		X		X
5.2.7					
5.2.8			X		X

Table 1: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM??
R1	
R2	
R3	
R4	
R5	X
R6	X
NFR1	X
NFR2	
NFR3	
NFR4	
NFR5	
NFR6	

Table 2: Traceability Matrix Showing the Connections Between Requirements and Instance Models

	A1	A2	A3	A4
TM??				
TM2				
GD3	X	X	X	X
DD1				
IM??				
LC1				X
LC2				
LC3				
LC4				
LC5				
LC6			X	
LC7				
LC8				
LC9				
LC10				
LC11				
LC12				
UC1	X			

Table 3: Traceability Matrix Showing the Connections Between Assumptions and Other Items

10 Values of Auxiliary Constants

MAINTAIN_FRAC = 1/10

References

[1] Wikipedia Contributors, "Natural number," Feb. 2023. Page Version ID: 1140714081.

[2] The Editors of Encyclopaedia Britannica, "real number," Oct. 2022.

[3] Wikipedia Contributors, "Positive real numbers," Jan. 2023. Page Version ID: 1130850074.

[4] D. O'Sullivan and D. J. Unwin, "Appendix A: Notation, Matrices, and Matrix Mathematics," in *Geographic Information Analysis*, pp. 373–394, Hoboken, NJ, USA: Wiley, 2nd ed., Mar. 2010.

[5] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Johns Hopkins University Press, 3rd ed., 2013.

[6] E. W. Weisstein, "Zero Matrix," 2023. Publisher: Wolfram Research, Inc.

[7] M. Taboga, "Augmented matrix," 2021.

[8] W. M. Farmer, *Simple Type Theory*. Computer Science Foundations and Applied Logic, Switzerland: Birkhäuser Cham, 1st ed., Jan. 2023.

[9] L. Lund and Anoka-Ramsey Community College, *Introduction to Chemistry*. Cambridge and Coon Rapids, MN, USA: LibreTexts, Jan. 2023.

[10] D. L. Parnas and P. Clements, "A rational design process: How and why to fake it," *IEEE Transactions on Software Engineering*, vol. 12, no. 2, pp. 251–257, February 1986.

[11] W. S. Smith, "capTemplate." https://github.com/smiths/capTemplate, Sept. 2022.

[12] W. S. Smith and L. Lai, "A new requirements template for scientific computing," in *Proceedings of the First International Workshop on Situational Requirements Engineering Processes – Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05* (J. Ralyté, P. Ågerfalk, and N. Kraiem, eds.), (Paris, France), pp. 107–121, In conjunction with 13th IEEE International Requirements Engineering Conference, 2005.

[13] W. S. Smith, L. Lai, and R. Khedri, "Requirements analysis for engineering computation: A systematic approach for improving software reliability," *Reliable Computing, Special Issue on Reliable Engineering Computation*, vol. 13, pp. 83–107, February 2007.

[14] N. Koothoor, *A Document Driven Approach to Certifying Scientific Computing Software*. MASc thesis, McMaster University, Hamilton, ON, Canada, May 2013.

[15] J. Carette, S. Smith, J. Balaci, T.-Y. Wu, S. Crawford, D. Chen, D. Szymczak, B. MacLachlan, D. Scime, and M. Niazi, "Drasil," Feb. 2021.

[16] B. MacLachlan, *A Design Language for Scientific Computing Software in Drasil*. MASc thesis, McMaster University, Hamilton, ON, Canada, June 2020.

[17] D. Chen, *Solving Higher-Order ODEs in Drasil*. MEng thesis, McMaster University, Hamilton, ON, Canada, Sept. 2022.

[18] R. M. Hartshorn, K.-H. Hellwich, A. Yerin, T. Damhus, and A. T. Hutton, "Brief guide to the nomenclature of inorganic chemistry," *Pure and Applied Chemistry*, vol. 87, no. 9-10, pp. 1039–1049, 2015.

[19] I. Hamid, "Balancing Chemical Equations by Systems of Linear Equations," *Applied Mathematics*, vol. 10, pp. 521–526, July 2019.

[20] HarperCollins Publishers, "hydrate."

[21] The Editors of Encyclopaedia Britannica, "nonstoichiometric compound," Oct. 2010.

[22] Wikipedia Contributors, "Integer programming," Mar. 2023. Page Version ID: 1142579898.

[23] Wikipedia Contributors, "List of chemical elements." https://en.wikipedia.org/wiki/List_of_chemical_elements, Jan. 2023. Page Version ID: 1130848496.

[24] W. S. Smith, "Assignment 2," Feb. 2020.

[25] "Balance Chemical Equation - Online Balancer," 2023.

[26] B. Risteski, "A new complex vector method for balancing chemical equations," *Materiali in Tehnologije*, vol. 44, pp. 193–203, 07 2010.