# System Verification and Validation Plan for ChemCode

Samuel Crawford

February 9, 2023

## 1 Revision History

| Date | Version | Notes |
|---|---|---|
| Feb. 5, 2023 | 0.0 | Create document and remove inapplicable content |
| Feb. 7-8, 2023 | 0.1 | Add input tests |
| Feb. 8, 2023 | 0.1.1 | Improve referencing of tests |
| | 0.1.2 | Add matrix conversion tests and improve input tests, including rationale, labelling, and chemical equations |
| Feb. 9, 2023 | 0.1.3 | Add tests for trivial equation |
| | 0.1.4 | Add feasibility tests |

## Contents

## List of Tables

[Remove this section if it isn’t needed —SS]

## List of Figures

[Remove this section if it isn’t needed —SS]

## 2 Symbols, Abbreviations and Acronyms

| symbol | description |
|---|---|
| T | Test |

[symbols, abbreviations or acronyms — you can simply reference the SRS [1] tables, if appropriate —SS]

[Remove this section if it isn't needed —SS]

This document ... [provide an introductory blurb and roadmap of the Verification and Validation plan —SS]

# 3 General Information

## 3.1 Summary

[Say what software is being tested. Give its name and a brief overview of its general functions. —SS]

## 3.2 Objectives

[State what is intended to be accomplished. The objective will be around the qualities that are most important for your project. You might have something like: "build confidence in the software correctness," "demonstrate adequate usability." etc. You won't list all of the qualities, just those that are most important. —SS]

## 3.3 Relevant Documentation

[Reference relevant documentation. This will definitely include your SRS and your other project documents (design documents, like MG, MIS, etc). You can include these even before they are written, since by the time the project is done, they will be written. —SS]

[1]

# 4 Plan

[Introduce this section. You can provide a roadmap of the sections to come. —SS]

## 4.1 Verification and Validation Team

[Your teammates. Maybe your supervisor. You shoud do more than list names. You should say what each person's role is for the project's verification. A table is a good way to summarize this information. —SS]

## 4.2 SRS Verification Plan

[List any approaches you intend to use for SRS verification. This may include ad hoc feedback from reviewers, like your classmates, or you may plan for something more rigorous/systematic. —SS]

[Maybe create an SRS checklist? —SS]

## 4.3 Design Verification Plan

[Plans for design verification —SS]

[The review will include reviews by your classmates —SS]

[Create a checklists? —SS]

## 4.4 Verification and Validation Plan Verification Plan

[The verification and validation plan is an artifact that should also be verified. —SS]

[The review will include reviews by your classmates —SS]

[Create a checklists? —SS]

## 4.5 Implementation Verification Plan

[You should at least point to the tests listed in this document and the unit testing plan. —SS]

[In this section you would also give any details of any plans for static verification of the implementation. Potential techniques include code walk-throughs, code inspection, static analyzers, etc. —SS]

## 4.6 Automated Testing and Verification Tools

[What tools are you using for automated testing. Likely a unit testing framework and maybe a profiling tool, like ValGrind. Other possible tools include a static analyzer, make, continuous integration tools, test coverage tools, etc. Explain your plans for summarizing code coverage metrics. Linters are another important class of tools. For the programming language you select, you should look at the available linters. There may also be tools that verify that coding standards have been respected, like flake9 for Python. —SS]

[If you have already done this in the development plan, you can point to that document. —SS]

[The details of this section will likely evolve as you get closer to the implementation. —SS]

### 4.7 Software Validation Plan

[If there is any external data that can be used for validation, you should point to it here. If there are no plans for validation, you should state that here. —SS]

[You might want to use review sessions with the stakeholder to check that the requirements document captures the right requirements. Maybe task based inspection? —SS]

[This section might reference back to the SRS verification section. —SS]

## 5 System Test Description

### 5.1 Tests for Functional Requirements

The tests in each section are given in order of increasing complexity/likelihood of the situation arising during use of ChemCode.

[Include a blurb here to explain why the subsections below cover the requirements. References to the SRS would be good here. —SS]

#### 5.1.1 Input Testing

In order for ChemCode to be useful, it needs to be able to receive a chemical equation from the user and store it to balance it later. This section defines tests for inputting chemical equations from R1 of the SRS [Add link —SC]. [Justify the choice of these specific tests. —SC]

T1: **Test for Small Valid Input**

Control: Manual

Initial State: ChemCode is started.

Input: A representation of the equation $O_2 \rightarrow O_3$ [2].

Output: The inputted chemical equation is stored in ChemCode.

Test Case Derivation: The inputted chemical equation is valid and trivial.

How test will be performed: A debug statement will be added to display the stored chemical equation and this representation will be manually compared to the given input.

T2: **Test for Valid Input**

Control: Manual

Initial State: ChemCode is started.

Input: A representation of the equation $C_2H_6 + O_2 \rightarrow CO_2 + H_2O$ [3].

Output: The inputted chemical equation is stored in ChemCode.

Test Case Derivation: The inputted chemical equation is valid and relatively small, but larger than the trivial one from T1.

How test will be performed: A debug statement will be added to display the stored chemical equation and this representation will be manually compared to the given input.

T3: **Test for Large Valid Input**

Control: Manual

Initial State: ChemCode is started.

Input: A representation of the following equation from [4].

$$KMnO_4 + HCl \rightarrow MnCl_2 + KCl + Cl_2 + H_2O$$

Output: The inputted chemical equation is stored in ChemCode.

Test Case Derivation: The inputted chemical equation is valid and larger than the one from T2.

How test will be performed: A debug statement will be added to display the stored chemical equation and this representation will be manually compared to the given input.

T4: **Test for Valid Input that is Infeasible due to Overconstrained System**

Control: Manual

Initial State: ChemCode is started.

Input: A representation of the following equation (modified from [3]).

$$K_4FeC_6N_6 + K_2S_2O_3 \rightarrow CO_2 + K_2SO_4 + NO_2 + FeS$$

Output: The inputted chemical equation is stored in ChemCode.

Test Case Derivation: The inputted chemical equation is infeasible since each compound has more than one element, so changing any coefficient affects the number of some other element, causing a chain reaction that does not converge. There is no solution to this system other than the trivial solution ($\mathbf{0}$) [3].

How test will be performed: A debug statement will be added to display the stored chemical equation and this representation will be manually compared to the given input.

T5: **Test for Valid Input that is Infeasible due to Conservation of Mass Violation**

Control: Manual

Initial State: ChemCode is started.

Input: A representation of the equation $C_2H_6 \rightarrow CO_2 + H_2O$.

Output: The inputted chemical equation is stored in ChemCode.

Test Case Derivation: The inputted chemical equation is infeasible since every element does not exist on both sides of the equation, which violates the Law of Conservation of Mass (TM1 from SRS [add link —SC]).

How test will be performed: A debug statement will be added to display the stored chemical equation and this representation will be manually compared to the given input.

T6: **Test for Valid Input with Nonstoichiometric Compound**

Control: Manual

Initial State: ChemCode is started.

Input: A representation of the equation $Fe_{0.95}O + O_2 \rightarrow Fe_2O_3$ [5].

Output: The inputted chemical equation is stored in ChemCode.

Test Case Derivation: The inputted chemical equation contains a non-stoichiometric compound (i.e., one with a fractional subscript).

How test will be performed: A debug statement will be added to display the stored chemical equation and this representation will be manually compared to the given input.

### 5.1.2 Matrix Conversion Testing

To solve a system of linear equations, ChemCode must first convert the inputted chemical equation into matrix form. This section defines tests for the matrix form conversion of chemical equations' stored representations from R2 of the SRS [Add link —SC]. [Justify the choice of these specific tests. —SC] Note that since swapping rows in a matrix doesn't change its solution, the matrices outputted may have their rows in a different order.

T7: **Test for Converting Small Valid Chemical Equation**

Control: Automated

Initial State: ChemCode is started with a representation of the equation $O_2 \rightarrow O_3$ [2] stored in memory.

Input: –

Output: The matrix $\begin{bmatrix} 2 & -3 \end{bmatrix}$.

Test Case Derivation: The stored chemical equation is valid and trivial.

How test will be performed: The outputted matrix form of the stored chemical equation will be automatically compared to the expected output.

T8: **Test for Converting Valid Chemical Equation**

Control: Automated

Initial State: ChemCode is started with a representation of the equation $C_2H_6 + O_2 \rightarrow CO_2 + H_2O$ [3] stored in memory.

Input: –

Output: Some matrix with the rows of the following matrix from [3], potentially in a different order:

$$\begin{bmatrix} 2 & 0 & -1 & 0 \\ 6 & 0 & 0 & -2 \\ 0 & 2 & -2 & -1 \end{bmatrix}$$

Test Case Derivation: The stored chemical equation is valid and relatively small, but larger than the trivial one from T7.

How test will be performed: The outputted matrix form of the stored chemical equation will be automatically compared to the expected output.

T9: **Test for Converting Large Valid Chemical Equation**

Control: Automated

Initial State: ChemCode is started with a representation of the equation $KMnO_4 + HCl \rightarrow MnCl_2 + KCl + Cl_2 + H_2O$ [4] stored in memory.

Input: –

Output: Some matrix with the rows of the following matrix, potentially in a different order:

$$\begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & -2 \\ 0 & 1 & -2 & -1 & -2 & 0 \end{bmatrix}$$

Test Case Derivation: The stored chemical equation is valid and larger than the one from T8.

How test will be performed: The outputted matrix form of the stored chemical equation will be automatically compared to the expected output.

T10: **Test for Converting Valid Chemical Equation that is Infeasible due to Overconstrained System**

Control: Automated

Initial State: ChemCode is started with a representation of the equation $K_4FeC_6N_6 + K_2S_2O_3 \rightarrow CO_2 + K_2SO_4 + NO_2 + FeS$ [3] stored in memory.

Input: –

Output: Some matrix with the rows of the following matrix[1], potentially in a different order:

$$\begin{bmatrix} 4 & 2 & 0 & -2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 \\ 6 & 0 & -1 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & -1 & 0 \\ 0 & 2 & 0 & -1 & 0 & -1 \\ 0 & 3 & -2 & -4 & -2 & 0 \end{bmatrix}$$

Test Case Derivation: The stored chemical equation is infeasible since each compound has more than one element, so changing any coefficient affects the number of some other element, causing a chain reaction that does not converge. There is no solution to this system other than the trivial solution ($\mathbf{0}$) [3].

How test will be performed: The outputted matrix form of the stored chemical equation will be automatically compared to the expected output.

T11: **Test for Converting Valid Chemical Equation that is Infeasible due to Conservation of Mass Violation**

Control: Automated

Initial State: ChemCode is started with a representation of the equation $C_2H_6 \rightarrow CO_2 + H_2O$ stored in memory.

Input: –

[1]While [3] does not include a matrix representation, the system of equations it provides has a typo: the equation for N should be $6x_1 = x_5$, since $NO_2$ only has one N.

Output: Some matrix with the rows of the following matrix, potentially in a different order:

$$\begin{bmatrix} 2 & -1 & 0 \\ 6 & 0 & -2 \\ 0 & -2 & -1 \end{bmatrix}$$

Test Case Derivation: The stored chemical equation is infeasible since every element does not exist on both sides of the equation, which violates the Law of Conservation of Mass (TM1 from SRS [add link —SC]).

How test will be performed: The outputted matrix form of the stored chemical equation will be automatically compared to the expected output.

T12: **Test for Converting Valid Chemical Equation with Nonstoichiometric Compound**

Control: Automated

Initial State: ChemCode is started with a representation of the equation $Fe_{0.95}O + O_2 \rightarrow Fe_2O_3$ [5] stored in memory.

Input: –

Output: Some matrix with the rows of the following matrix, potentially in a different order:

$$\begin{bmatrix} 0.95 & 0 & -2 \\ 1 & 2 & -3 \end{bmatrix}$$

Test Case Derivation: The stored chemical equation contains a nonstoichiometric compound (i.e., one with a fractional subscript).

How test will be performed: The outputted matrix form of the stored chemical equation will be automatically compared to the expected output.

### 5.1.3 Feasibility Checking Testing

To balance a chemical equation, the equation must be able to be balanced. This section defines tests for determining if a chemical equation is feasible from R3 of the SRS [Add link —SC]. [Justify the choice of these specific tests. —SC] Note that since swapping rows in a matrix doesn't change its solution, the matrices outputted may have their rows in a different order.

T13: **Test for Feasibility of Small Valid Chemical Equation**

Control: Automated

Initial State: ChemCode is started.

Input: $\begin{bmatrix} 2 & -3 \end{bmatrix}$.

Output: T.

Test Case Derivation: The matrix represents a chemical equation that is valid and trivial.

How test will be performed: The outputted Boolean will be automatically compared to the expected output.

T14: **Test for Feasibility of Valid Chemical Equation**

Control: Automated

Initial State: ChemCode is started.

Input: Some matrix with the rows of the following matrix from [3], potentially in a different order:

$$\begin{bmatrix} 2 & 0 & -1 & 0 \\ 6 & 0 & 0 & -2 \\ 0 & 2 & -2 & -1 \end{bmatrix}$$

Output: T.

Test Case Derivation: The matrix represents a chemical equation that is valid and relatively small, but larger than the trivial one from T13.

How test will be performed: The outputted Boolean will be automatically compared to the expected output.

T15: **Test for Feasibility of Large Valid Chemical Equation**

Control: Automated

Initial State: ChemCode is started.

Input: Some matrix with the rows of the following matrix, potentially in a different order:

$$\begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & -2 \\ 0 & 1 & -2 & -1 & -2 & 0 \end{bmatrix}$$

Output: T

Test Case Derivation: The matrix represents a chemical equation that is valid and larger than the one from T14.

How test will be performed: The outputted Boolean will be automatically compared to the expected output.

T16: **Test for Feasibility of Valid Chemical Equation that is Infeasible due to Overconstrained System**

Control: Automated

Initial State: ChemCode is started.

Input: Some matrix with the rows of the following matrix[2], potentially in a different order:

$$\begin{bmatrix} 4 & 2 & 0 & -2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 \\ 6 & 0 & -1 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & -1 & 0 \\ 0 & 2 & 0 & -1 & 0 & -1 \\ 0 & 3 & -2 & -4 & -2 & 0 \end{bmatrix}$$

Output: F.

Test Case Derivation: The matrix represents a chemical equation that is infeasible since each compound has more than one element, so changing

[2]While [3] does not include a matrix representation, the system of equations it provides has a typo: the equation for N should be $6x_1 = x_5$, since $NO_2$ only has one N.

any coefficient affects the number of some other element, causing a chain reaction that does not converge. There is no solution to this system other than the trivial solution ($\mathbf{0}$) [3].

How test will be performed: The outputted Boolean will be automatically compared to the expected output.

T17: **Test for Feasibility of Valid Chemical Equation that is Infeasible due to Conservation of Mass Violation**

Control: Automated

Initial State: ChemCode is started.

Input: Some matrix with the rows of the following matrix, potentially in a different order:

$$\begin{bmatrix} 2 & -1 & 0 \\ 6 & 0 & -2 \\ 0 & -2 & -1 \end{bmatrix}$$

Output: F.

Test Case Derivation: The matrix represents a chemical equation is infeasible since every element does not exist on both sides of the equation, which violates the Law of Conservation of Mass (TM1 from SRS [add link —SC]).

How test will be performed: The outputted Boolean will be automatically compared to the expected output.

T18: **Test for Feasibility of Valid Chemical Equation with Nonstoichiometric Compound**

Control: Automated

Initial State: ChemCode is started.

Input: Some matrix with the rows of the following matrix, potentially in a different order:

$$\begin{bmatrix} 0.95 & 0 & -2 \\ 1 & 2 & -3 \end{bmatrix}$$

Output: T.

Test Case Derivation: The matrix represents a chemical equation that contains a nonstoichiometric compound (i.e., one with a fractional subscript).

How test will be performed: The outputted Boolean will be automatically compared to the expected output.

## 5.2 Tests for Nonfunctional Requirements

[The nonfunctional requirements for accuracy will likely just reference the appropriate functional tests from above. The test cases should mention reporting the relative error for these tests. Not all projects will necessarily have nonfunctional requirements related to accuracy —SS]

[Tests related to usability could include conducting a usability test and survey. The survey will be in the Appendix. —SS]

[Static tests, review, inspections, and walkthroughs, will not follow the format for the tests given below. —SS]

### 5.2.1 Area of Testing1

**Title for Test**

T19: **Title of Test**

Type: Functional, Dynamic, Manual, Static etc.

Initial State:

Input/Condition:

Output/Result:

How test will be performed:

T20: **Title of Test**

Type: Functional, Dynamic, Manual, Static etc.

Initial State:

Input:

Output:

How test will be performed:

#### 5.2.2 Area of Testing2

...

### 5.3 Traceability Between Test Cases and Requirements

[Provide a table that shows which test cases are supporting which requirements. —SS]

## 6 Unit Test Description

[Reference your MIS (detailed design document) and explain your overall philosophy for test case selection. —SS] [This section should not be filled in until after the MIS (detailed design document) has been completed. —SS]

### 6.1 Unit Testing Scope

[What modules are outside of the scope. If there are modules that are developed by someone else, then you would say here if you aren't planning on verifying them. There may also be modules that are part of your software, but have a lower priority for verification than others. If this is the case, explain your rationale for the ranking of module importance. —SS]

### 6.2 Tests for Functional Requirements

[Most of the verification will be through automated unit testing. If appropriate specific modules can be verified by a non-testing based technique. That can also be documented in this section. —SS]

#### 6.2.1 Module 1

[Include a blurb here to explain why the subsections below cover the module. References to the MIS would be good. You will want tests from a black box perspective and from a white box perspective. Explain to the reader how the tests were selected. —SS]

1. test-id1

Type: [Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic —SS]

Initial State:

Input:

Output: [The expected result for the given inputs —SS]

Test Case Derivation: [Justify the expected value given in the Output field —SS]

How test will be performed:

2. test-id2

   Type: [Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic —SS]

   Initial State:

   Input:

   Output: [The expected result for the given inputs —SS]

   Test Case Derivation: [Justify the expected value given in the Output field —SS]

   How test will be performed:

3. ...

#### 6.2.2 Module 2

...

### 6.3 Tests for Nonfunctional Requirements

[If there is a module that needs to be independently assessed for performance, those test cases can go here. In some projects, planning for nonfunctional tests of units will not be that relevant. —SS]

[These tests may involve collecting performance data from previously mentioned functional tests. —SS]

#### 6.3.1 Module ?

1. test-id1

   Type: [Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic —SS]

   Initial State:

   Input/Condition:

   Output/Result:

   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

   How test will be performed:

#### 6.3.2 Module ?

...

### 6.4 Traceability Between Test Cases and Modules

[Provide evidence that all of the modules have been considered. —SS]

## References

[1] A. Author, "System requirements specification." https://github.com/..., 2019.

[2] D. W. Fahey and M. I. C. L. A. Hegglin, "Twenty Questions and Answers About the Ozone Layer: 2010 Update, Scientific Assessment of Ozone Depletion: 2010," tech. rep., World Meteorological Organization, Geneva, Switzerland, Mar. 2011.

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

[4] S. Taylor, “Balancing Complex Chemical Equations with Chemical Equations Examples,” June 2021.

[5] Doubtnut, “When non stoiciometric compound Fe0.95O is heated in presence of oxygen then it convents into Fe2O3. Which of the following statement is correct?.”

# 7 Appendix

This is where you can place additional information.

## 7.1 Symbolic Parameters

The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

## 7.2 Usability Survey Questions?

[This is a section that would be appropriate for some projects. —SS]