

# Tech Specification - Circuit Simulator

Adam Rehman, Brandon Cann, Xin Wang <sup>1</sup>

May 15, 2020

<sup>1</sup>document compiled by Xin Wang

# Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
1.1	Product Requirements . . . . .	2
1.2	Assumptions . . . . .	2
1.3	Out of scope . . . . .	2
1.4	Open Questions . . . . .	2
<b>2</b>	<b>Approach</b>	<b>3</b>
2.1	Program flowchart . . . . .	3
2.2	Git management . . . . .	3
2.3	Milestones . . . . .	3

# 1 Overview

Please see document: *ELEC40006 Specification Q3* for information provided to the team.

## 1.1 Product Requirements

- Program must be able to read a SPICE netlist which the user inputs.
  - Program should recognise supported circuit elements e.g. R, L and C
  - Program should recognise and extract parameters for *.tran* function
- Program must perform *Transient Analysis* based on the netlist provided.
- Program must output result of simulation in a *.csv* format.

## 1.2 Assumptions

- User input adheres to the SPICE netlist format
- Netlist contains only the following elements:
  - Resistor
  - Capacitor
  - Inductor
  - Independent sources
  - Dependent sources

## 1.3 Out of scope

- This program applies only to transient analysis based on input provided by the user. It does not support AC analysis.
- Complex components such as BJTs and MOSFETs are not supported.
- A user interface is not required, program executes when input file is inputted.

## 1.4 Open Questions

- What the data structure to store the circuit should be?
- Does the program check if circuit input is correct?
- Is there any efficiency expectations?

- Program should be written in C++? Python is so much more programmer-friendly.

## 2 Approach

### 2.1 Program flowchart

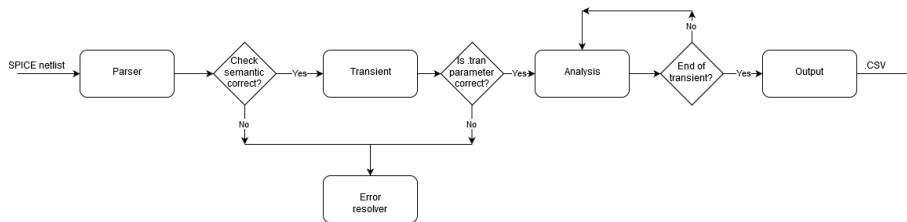


Figure 1: Program flowchart

### 2.2 Git management

The feature each teammate is responsible for will be implemented in a feature branch until it is fully tested and other teammates are familiarised with its interface.

Xin Wang will be in charge of Git repo management responsibilities.

### 2.3 Milestones

- v1.0:
  - Basic nodal analysis possible with resistor and independent sources.
  - **Transient not possible yet, only calculates one instance in time.**
- v2.0:
  - Transient analysis ability implemented.
  - Dependent sources supported.
- v3.0:
  - Capacitor and Inductor supported.
- Codebase optimisation.
- v4.0:
  - Non-linear components supported.
- Codebase optimisation.