XycLOps Vision

Table of Contents

Document Purpose

This document seeks to outline the developers' thoughts on future improvements that could be made to XycLOps. These observations are drawn from experience built during tool development and use. Future contributors to the XycLOps tool are encouraged to draw inspiration from the following to make meaningful and impactful contributions.

List

- ### Extraneous File I/O Reduction
    - Currently, every Xyce iteration outputs a file that is in turn parsed into memory for XycLOps's data operations. This means every Xyce iteration has two associated file I/O operations. Since I/O bound operations are notoriously slow, finding a way to eliminate this would most likely improve tool performance. (Maybe something like writing files to /dev/shm or memory mapped files)

- ### Parallelism
    - Currently, all Xyce iterations are run serially. Finding a way to run Xyce iterations in a parallel fashion, perhaps even utilizing GPUsm would certainly increase tool performance. Note: This is hard and could require major optimization loop changes since there does not appear to be a trivial way to parallelize SciPy least_squares.

- ## Multi-platform Support
    - Currently, this tool is tailored to support a Windows experience. While other platforms do work, keeping broad multi-platform support in mind throughout the future development process would improve the tool's versatility and reach.
- ## Robust User Experience
    - Currently, Xyce offers few user experience features such as integrated explanations of key components and user input checks. Adding such features would make the tool more robust and resistant to unskilled usage.
- ## Non-Transient Analyses
    - Currently, the tool only supports transient analysis using Xyce. Increasing the tool's scope to support different analysis types that Xyce can use, especially AC analysis, would greatly increase the utility of this tool.
- ## Optimization Settings
    - Currently, the tool uses SciPy's least_squares function for optimization workload with many key parameters hardcoded. Making these parameters customizable by the user (with proper explanation) would make the tool much more responsive to the diverse use cases that entail different computational rigor for the optimization engine.
- ## Model Compatibility
    - Currently, XycLOps can only work with certain circuit models. Making the tool compatible with common circuit models and libraries (e.g. LTSpice default models) would greatly increase the tool's reach and give users more freedom in their choice of schematic editor.
- ## Non-curvefit Optimization
    - Currently, XycLOps only accepts specified voltage curve input as an optimization criteria. Adding more types of optimization criteria could increase applicable use cases.
- ## Schematic Editor Integration
    - Currently, XycLOps is only loosely coupled with a schematic editor, relying on only the netlist files that are generated by the editor. More closely integrating XycLOps with a schematic editor, graphically and in terms of error handling, could greatly improve user experience.
- ## Multiple Optimizations
    - Currently, the parameter setup work for XycLOps allows a user to run only one optimization. Changing this to allow multiple optimizations with varying tran commands would allow users to do more with less setup time.

- ## Xyce Error Visibility
    - Currently, XycLOps assumes that user netlists will run perfectly on Xyce. Giving the tool the ability to correct basic Xyce errors such as unit formatting and displaying specific Xyce errors to users would make the tool much easier to use for Xyce novices.