

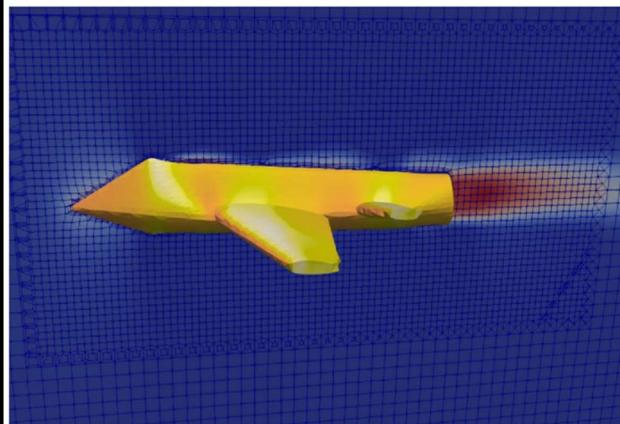
(第1回)ngspiceのpython操作方法検討 他

1 Dec 2025

AirtomoR

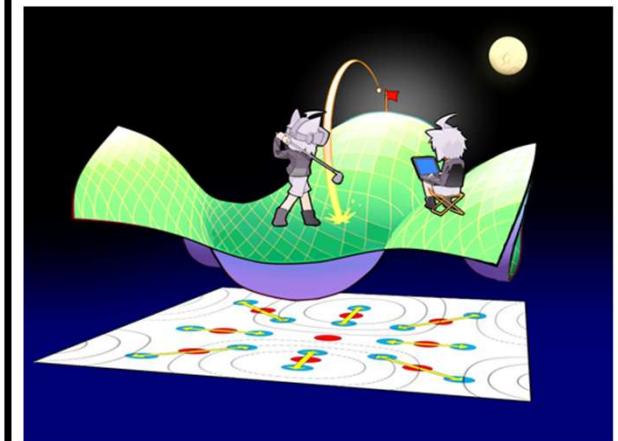
Kindle unlimited
で読んでね

3D 物理シミュレーションで実践できる
多パラメータ最適化の新手法
(アップデート版)
パラメータ分配スコア - パラメータ分析ユーティリティソフト (GUI,python3-pyside2)
3D 形状最適化シミュレーションモデル 有



3年前位

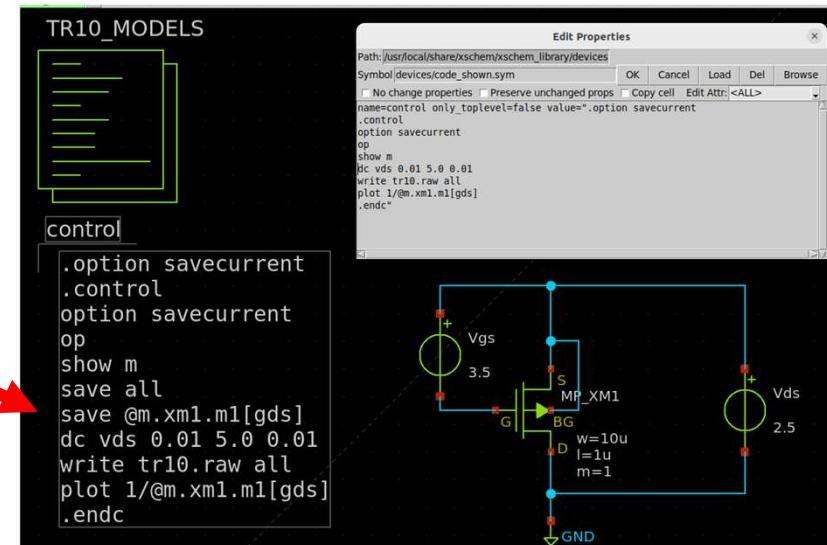
グリッドサーチ
多目的値多パラメータ最適化
ソフトウェアの拡張機能解説と
物理シミュレーション例示



今年5月

主にGrokに確認しても.rawファイルすら出力できず

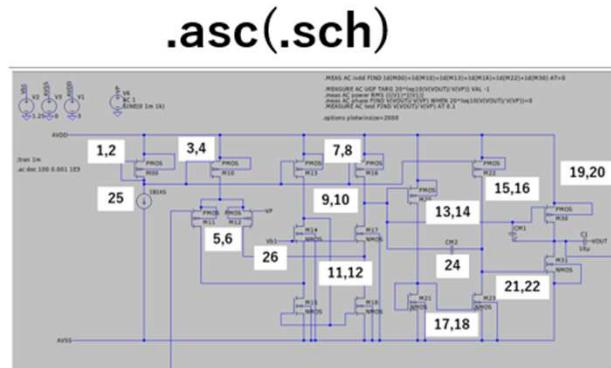
writeを入れても
rawファイルが出力されない



・回路のdirective(LTSpice的な表現?)にwriteを入れれば
出力とのことのはずだが出ない。
(saveというのが上手く機能してないかも)
xschemに慣れている方なら想定がつくと思いますので御教えください。

(今日として画面共有で実演して頂く。ファイルもDiscordにup願。
それを見せて私の方でもトライしてみる)

xschem段階でのパラメータ更新は困難そう



変数編集コード

```
with open('./temp/amp3.net',encoding='CP932') as f:
    s = f.read()
    lines = s.split("\n")
    lines[8] = lines[8] + " " + "m0" + " " + str(param_list[0]) + " " + "l0" + " " + str(param_list[1])
    lines[9] = lines[9] + " " + "m0" + " " + str(param_list[2]) + " " + "l0" + " " + str(param_list[3]) + " " + "m0" + " " + str(param_list[4])
    lines[10] = lines[10] + " " + "m0" + " " + str(param_list[5]) + " " + "l0" + " " + str(param_list[6]) + " " + "m0" + " " + str(param_list[7])
    lines[11] = lines[11] + " " + "m0" + " " + str(param_list[8]) + " " + "l0" + " " + str(param_list[9])
    lines[12] = lines[12] + " " + "m0" + " " + str(param_list[10]) + " " + "l0" + " " + str(param_list[11]) + " " + "m0" + " " + str(param_list[12])
    lines[13] = lines[13] + " " + "m0" + " " + str(param_list[13]) + " " + "l0" + " " + str(param_list[14]) + " " + "m0" + " " + str(param_list[15])
    lines[14] = lines[14] + " " + "m0" + " " + str(param_list[16]) + " " + "l0" + " " + str(param_list[17]) + " " + "m0" + " " + str(param_list[18])
    lines[15] = lines[15] + " " + "m0" + " " + str(param_list[19]) + " " + "l0" + " " + str(param_list[20]) + " " + "m0" + " " + str(param_list[21])
    lines[16] = lines[16] + " " + "m0" + " " + str(param_list[22]) + " " + "l0" + " " + str(param_list[23]) + " " + "m0" + " " + str(param_list[24])
    lines[17] = lines[17] + " " + "m0" + " " + str(param_list[25]) + " " + "l0" + " " + str(param_list[26]) + " " + "m0" + " " + str(param_list[27])
```

.net(.cir)

```
IBIAS N001 AVSS
CM2 N007 N002 C
CM1 N002 N006 C
V1 AVDD 0 3
V2 Vb1 0 1.25
V3 AVSS 0 0
V6 VP 0 SINE(0 1m 1k) AC 1
M00 N001 N001 AVDD AVDD PMOS
M10 N003 N001 AVDD AVDD PMOS
M11 N009 N004 N003 N003 PMOS
M12 N008 VP N003 N003 PMOS
M13 N005 N001 AVDD AVDD PMOS
M16 N002 N001 AVDD AVDD PMOS
M14 N005 Vb1 N009 AVSS NMOS
M15 N009 N005 AVSS AVSS NMOS
M17 N002 Vb1 N008 AVSS NMOS
M18 N008 N005 AVSS AVSS NMOS
```

.log(.raw経由からも可)

```
test: v(vout)/v(vp)=(105.86dB,-3.84881°) at 0.1
phase: v(vout)/v(vp)=(0.000558095dB,-94.7432°) at 286041
power: RMS(i(v1))=0.000256367 FROM 0.001 TO 1e+009
ugf=319960 FROM 0.001 TO 319960
ivdd: i(v1)=(-20.9287dB,-3.84987°) at 0.1
```

```
with open('./temp/amp3_1.log',encoding='CP932') as f:
    s = f.read()
    lines = s.split("\n")
    test_result_list1 = []
    print(lines[7],lines[8])
    for i in range(len(lines)):
        if lines[i][0:6] == "power":
            print("power found")
            test_result_list1=re.findall(pattern, lines[i])
            print(test_result_list1)

    test_result1 = test_result_list1[1]
    #test_result1=1000.0*(10*(0.05*float(test_result1)))
    test_result1=1000*float(test_result1)
    test_result1=str(test_result1)
```

スコア取得コード

LTSpiceならspicelibで左のようなGUIモデルとして素子値にアクセス化(但し一部未実装)
xschemはどうやらできなさそう。KiCADだとpython対応可能らしい(KiCAD公式で対応化?)

Grokに確認したところngspiceのGUIラッパーとして
KiCADなら直接操作できる可能性が高いらしいが
KiCADはその他ツールとの連携具合がわからぬので避けた方が良い?

Netlist操作で妥協するとしてGrokでルーチン化可能

明日夜の勉強会でそのまま投げて使ってください！

5分で決まるように「選択肢+チェック欄」形式にしてあります。

【チーム名 / プロジェクト名】 _____

1. 対象プロセス（複数選択可）
 sky130A GF180MCU IHP SG13G2 TR10 (東海理化180nm BiCD) その他: _____

2. シミュレーション作業ディレクトリ
 xschem/ の直下に sim/ を作る（推奨）
 各回路ごとに sim_<回路名>/ を作る
 回路名+cornerで sim_<回路名>_<corner>/ を作る（例: sim_opamp_tt）

3. ファイル命名規則（絶対に変えない）
netlist → {回路名}.cir

4. xschem CLI コマンド（固定）
xschem -q -n -s --netlist_filename {cirファイル} {schファイル}

5. ngspice CLI コマンド（固定）
ngspice -b -r {rawファイル} -o {logファイル} {cirファイル}

6. Pythonスクリプトの必須構成（私が毎回この通りに書く）
【必須関数】

```
def run_simulation(sch_file: str | Path, corner: str = "tt") -> dict:
    # 戻り値は必ず dict で以下を含む
    return {
        "sch": Path,
        "cir": Path,
        "raw": Path,
        "log": Path,
        "work_dir": Path
    }
```

}

【必須インポート】

```
import subprocess
from pathlib import Path
```

【エラー処理】

失敗したら必ず raise RuntimeError(エラーメッセージの最後20行)

7. コーディングスタイル（私が厳守するもの）
・変数名は snake_case
・日本語コメント可（むしろ推奨）
・spicelib.RawRead で読む例を最後に必ず付ける
・バックアップは作らない（gitで管理する前提）

8. モデルファイルのinclude方法（チームで統一）
 .sch の一番下に .include "../models/tr10_tt.lib" のように書く
 xschemrc に全局includeを書く（上級者向け）

9. 追加で欲しい便利関数（チェックでON/OFF）
 自動で matplotlib プロットしてPNG保存
 corner一括実行 (tt/ss/ff/mc) を並列で回す
 W/L/m を引数で変えてスイープ実行
 測定値 (gm, id, vthなど) を自動抽出してcsv吐き出し

10. その他チーム特有ルール（自由記入）

Netlist段階でパラメータ更新は人間の見通しが悪いけど
LLMにやらせると決めれば特に悩みもないかも？(テンプ
レ化しておけば再現性のある出力をしてもらえるとの事)

紹介頂いたPLLの確認

- ・どなたかxschemで開いて性能指標値等が想定できれば御教えください。
(12/8にも皆様と少し議論して
来年初頭に最適化戦略をお見せしたい)

<https://github.com/atuchiya/DC23-LTC2/tree/japan-test/PLL/>