Joyce Huang
sh2393

Programming Assignment 3: Parameter Extraction

Introduction

The goal of this lab is to construct a program to perform parameter extraction from least square fitting. The model we used for parameter extraction is the EKV model described below.

$$I_{D,EKV} = \underbrace{I_S \log^2\left(1+\exp\left(\frac{\kappa(V_{GB}-V_{th})-V_{SB}}{2V_T}\right)\right)}_{I_F} - \underbrace{I_S \log^2\left(1+\exp\left(\frac{\kappa(V_{GB}-V_{th})-V_{DB}}{2V_T}\right)\right)}_{I_R}$$

For this nonlinear optimization, we used a quasi-Newton method for extracting parameters, and we observed the convergence of this iterative solution. The program also includes automatic validation. The data computed from the program is saved, but graphical visualization is done separately. as I wrote my program in C++ so there is no suitable library to perform plotting.

Implementation

Objects

1. Full Matrix Object

I constructed a 3 by 3 full matrix object to store the matrix type to help compute some intrinsic values of the matrix. To functions are implemented: compute determinant and compute inverse.

Methods

1. Direct Matrix Solver

The direct matrix solver is an extension from the matrix object. It is implemented by appending the solution vector at the last column, performs pivoting and alters the matrix.

2. Quasi-Newton Method

This module contains the iterative solution for parameter extraction. It performs calculation with the Hessian matrix. Then it implements the derivative calculations and nonlinear optimization searches.

3. Full gridded search

As required in task 6, the program also implements a full parameters search. The implementation searches the full gridded regions of I_S in (10^{-8}A, 3×10^{-8}A, 10^{-7}A, 3×10^{-7}A, 10^{-6}A, 3×10^{-6}A, 10^{-5}A, 3×10^{-5}A), κ in (0.5, 0.6, 0.7, 0.8, 0.9) and V_{th} in (0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0).

Validation

There are two testing and validation function. The matrix operation and the row pivot matrix solver is unit tested. Also, the parameter extraction program is validated with the power law. The program is implemented to perform a validation of your parameter extraction program for $y = c_0x^m$ (power law), where c_0 will be 10 and m will be –0.5.

Joyce Huang
sh2393

Results

We used the outputNMOS.txt measured data to perform parameter extraction.

Task 4: $S_{model} = I_D(V_{GS}, V_{DS}; I_S, \kappa, V_{th})$.
Task 5: $S_{model} = I_D(V_{GS}, V_{DS}; I_S, \kappa, V_{th})/I_{Dmeasured}$.

$$\|\Delta\|_2 = \frac{\Delta I_S^2}{I_S^2} + \frac{\Delta \kappa^2}{\kappa^2} + \frac{\Delta V_{th}^2}{V_{th}^2} \qquad \Delta S_{ai} = \frac{\dfrac{S(x_1, x_2 \ldots; a_1, \ldots a_i + \Delta a_i \ldots, a_m)}{S(x_1, x_2 \ldots; a_1, \ldots a_i \ldots, a_m)}}{\dfrac{a_i + \Delta a_i}{a_i}}$$

Quasi-Newton Methods

	I_S	κ	V_{th}	$\|V\|$	$\|\Delta\|$	ΔS_{IS}	ΔS_κ	ΔS_{Vth}
Task 4	9.99e-8	1.249	-3.358	1.178e-3	2.60	-0.59	-0.52	nan
Task 5	1.03e-7	1.02	1.00	2.41e6	6.48e-10	2.00	2.42	-22.51

Table above shows the various metric for iterative solution.