Gauss-Seidel Loadflow Program For the HP50g Calculator V1
Gerardo V. Lozada, M.S., P.E.E.

Introduction

A Gauss-Seidel loadflow program to calculate bus voltages and line flows in a power system written in C and cross-compiled using HPGCC. It requires the ARM Toolbox runtime system to be installed in your HP50g (or 49g+). Since it is written in C and compiled to run in native ARM processor instructions, it is pretty fast in spite of the numerous iterations normally required by the method. Although I am presently developing a Newton-Raphson algorithm version, the Gauss-Seidel version seems pretty fast enough, requires far less memory and is less prone to divergence. In some cases, convergence oscillation may occur as the mismatch becomes very small but this easily solved by pressing the ON button to terminate the iterative voltage computation and proceed to the line flow calculations. The installation requires an SD card in port 3 although the user can install it manually in in another port which requires modifying the UserRPL wrapper program GSLFW.

Input data is read from the stack, output data is likewise dumped to the stack.

Input Data Matrices:

Input data matrices are entered on the UserRPL stack or via the built-in Matrix Writer. Input records are in rows with each record's fields in columns.

Input Bus Data Fields (matrix column entries per row/record): Bus Number, Specified or Initial Voltage Magnitude (p.u.), Specified or Initial Phase Angle (Deg.), Real Generation (MW), Reactive Generation (MVAR), Minimum Reactive Generation (MVAR), Maximum Reactive Generation (MVAR), Real Load (MW), Reactive Load (MVAR) and Station Capacitor (MVAR). For unused fields, put a zero.

Input Line /Transformer Data Fields (matrix column entries per row/record): From Bus No., To Bus No., Resistance (p.u.), Reactance (p.u.), Line Charging (MVAR), Transformer High Side Tap (p.u.), Transformer High Side Minimum Tap (p.u.) and Transformer High Side Maximum Tap (p.u.). For unused fields, put a zero.

Output Data Matrix Strings:

Output records are in rows with each record's fields in columns.

Output Bus Data (matrix column fields per row/record): Bus Number, Voltage Magnitude (p.u.), Voltage Phase Angle (Deg.), Real Generation (MW), Reactive Generation (MVAR), Real Load (MW), Reactive Load (MVAR) and Station Capacitor Output (MVAR)

Output Line/Transformer Data (matrix column fields per row/record): From Bus No., To Bus No., Real Power Flow (MW), Reactive Power Flow (MVAR) and Transformer Tap. Plus (+) sign indicates power flow away from bus, negative sign indicates power flow towards the bus.

Installation

To install, transfer the UserRPL wrapper program gslfw.hp, the HPGCC-compiled C program gslf.hp, the UserRPL installer program install.hp, the runtime ARM toolbox setup.bin as well as sample data files bus2a.hp and line2a.hp which are on 100 MVA base with Bus No.1 as the swing bus. The installer program (taken from Egan Ford's *Extend Your 50g With C*) converts the compiled program into a stand-alone executable and transfers it to an EXTEND folder within the SD card (port 3). Please refer to Armtoolbox INSTALL.TXT for instructions on installing setup.bin on your calculator.

Program Usage

The user runs the program by invoking the wrapper program after entering the four arguments on the stack (Base MVA, Bus Input Data, Line Input Data and Swing Bus Number). The program reads the input data after pre-processing by the wrapper and runs. The compiled C program will not read the input data correctly unless it is invoked through the UserRPL wrapper program. The program pauses at certain intervals to give the user time to view the program's intermediate output; the user continues execution by pressing ENTER. In case the program oscillates during iteration close to convergence where the voltage mismatch is very close to the tolerance, the user has only to press the ON key to break out of the iteration cycle and proceed to line/transformer flow computation.

Program output data is left on the stack as strings which can be converted to numeric matrices using the UserRPL command obj-> from the PRG, TYPE submenu.

GVL/11-19-2009
gvlozada@yahoo.com