Using the TI-84 Sudoku Programs

by Mark Spahn, Ron Lancaster, Deborah Moore Russo and Gerald Rising

The instructions presented in this paper will show you how to solve sudoku puzzles with a TI-84 calculator. Although you should enter all 34 programs and subprograms in your calculator, you need only access two programs, AGETGRID and AASUDOKU, to solve these puzzles.

PROGRAM:AGETGRID

Running AGETGRID allows you to enter the numbers appearing in the cells of the sudoku you wish to solve. Using it is straightforward:

1. Open AGETGRID and press ENTER. This will produce X=? on your screen.

2. Respond to this by typing the numbers in the first row of sudoku cells, typing zeroes for any empty cells. For example, if the sudoku contains only a 3 in the 4th column and a 5 in the 7th column of row 1, you would type: 000300500. Alternatively, you can simply type: 300500, because you may omit initial zeroes.

3. Press ENTER and continue in this fashion until all nine rows are entered.

When you have pressed ENTER for the 9th row, a matrix will be displayed on your screen. Unless you have mistyped, this will show the numbers in the positions of your sudoku. (You will have to maneuver to the right and down to see the matrix rows and columns that do not fit on the screen display.) You will also see this same array displayed when you run the program with AASUDOKU. If you discover errors then, quit the program. In either case simply start over with AGETGRID.

At this point you will have entered your sudoku in both matrices [A] and [C]. Matrix [A] is copied to matrix [C] by the program so that you can use matrix operations to store [C] back in [A] without having to retype the cell values. This allows you to rerun the program if you should wish to do so.

You are now ready to solve this sudoku with the program AASUDOKU.

---

1To use a TI-83 calculator to solve the same puzzles, some program modifications must be made as the TI-83 does not include timing features and those must be removed from the code.

2Those programs are: AASUDOKU, AGETGRID, CELLHYPS, CELLS, CHKEY, DINRC, ENNEADS, ENNHYPS, ERASE8, ERASEJ, ERI, FILLCELL, HILITCEL, HILITENN, HYPPAIRS, INSsymbol, OSCAN, PRIMPROD, RCPIXTJ, REV1, SCAN, SETB, SETHYRCD, SETMTRXA, SETRCB, SETTWINH, SHCELL, SHGRID, SHHYP, SHPROG, SHTIME, SHZEROCT, UNWINDL2 and WAIT.

3These two programs each begin A so that they will be near the beginning of your program listing.
PROGRAM: AASUDOKU

Running AASUDOKU is even simpler than running AGETGRID, but there are some choices you should know about that allow you to make changes as the program is running.

To solve a sudoku that you used AGETGRID to enter in matrix [A], simply open AASUDOKU and press [ENTER]. The calculator will immediately perform a sequence of operations:

- It will start timing the solution.
- It will draw the sudoku array.
- It will fill in the given digits in the array.
- It will display the time used and the number of empty cells remaining.
- It will begin testing cells to find new digits to add to the array, displaying which cells or later enneads are being tested and what digits are being filled in.

You need do nothing more. The calculator will work until all the cells are filled. Then it will report the length of time it has taken to reach the solution. Additional information is also reported as the solution proceeds: which cell is being considered, which digit is being tested for which ennead, how many cells were filled by the latest pass through the two basic algorithms and, if hypotheses are necessary, what they are and which one is being tested.

How to Control Timing

There is, however, one thing that you can control as the program is running. The program is preset to a display delay of 2 seconds. When it has determined what digit will fill a given cell, it waits for 2 seconds to allow you to follow what is happening. You can change this delay to any number of seconds from 0 to 9, simply by holding down the corresponding digit key for several seconds – long enough for cell testing to complete a row, column or box.

Another choice you have is to make the program pause whenever it is about to make a change and wait for you to have it resume processing. To change to this mode, press the decimal point key, [•]. Each time the program stops in this mode, press [ENTER] to continue. To return to one of the timed delays, press one of the digit keys while the program is running.

You should find that these timing changes are simpler to use than they are to explain.

By following these instructions, you will be able to solve virtually all sudokus. Only in the case of the most difficult puzzles will it be necessary to do more. You will be notified that this is the case when your program displays, HYPS FAIL’D. TOO BAD.
A Final Refinement for Exceptionally Difficult Sudokus

If you are solving a very difficult sudoku and reach an impass, there is still another procedure you can implement. The subprogram ENNHyps has been turned off in subprogram HYPPAIRS. This has been done because identifying these additional hypotheses slows the processing of the large majority of puzzles that do not need them.

You can turn this subprogram on by changing the program line immediately proceeding prgmENNHYPS from If 0 to If 1. To save time, also change the line before prgmCELLHYPS from If 1 to If 0 to turn that subprogram off. (You will have already tested those hypotheses.) Once you have made the changes, simply restart AASUDOKU and it will continue solving from the state of the sudoku when your earlier algorithms failed. It will, however, repeat the pre-hypothesizing algorithms before generating and testing the ennead-related hypotheses.